Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Rong, CHEN

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M. Sc. (Systems Science)

GRADE - DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Multicriteria Approach for Risk-based Science Prioritization

D. Lane

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

S. Findlay

G. Kersten

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

University of Ottawa

Master's Program in Systems Science

Thesis

Multicriteria Approach for Risk-based Science Prioritization

Student Name: Rong Chen

Student No: 2986157

Thesis Director:

Professor Daniel E. Lane

Faculty of Graduate and Postdoctoral Studies

University of Ottawa

September 1, 2004

Your file Votre référence
ISBN: 0-494-01435-0
Our file Notre référence
ISBN: 0-494-01435-0

Acknowledgments

Several people have helped me finish this thesis research. It would be nice to have an opportunity to publicly thank them all.

First of all, I would like to mention my supervisor, Professor Daniel Lane, who dedicated plenty of his time to instructing the research and reviewing my thesis. Dr. Lane has devoted to the research on decision-making processes, simulation modeling and control of dynamic systems, especially in the area of fishery management. His research and reputation capture me at the very beginning of this Master program. And now, as I submit this thesis to complete the study, I am truly grateful for his tremendous help, and I am glad that I have this chance to express my appreciation.

This thesis research has been using the information provided by people from DFO. And I had this wonderful experience to present it in the DFO Science Maritimes Region workshop, which brought me very positive feedback and hence encouraged me most to finish the research. I hereby want to thank Sherry Niven and Nicole Asselin for providing valuable information, and the Director and Managers of DFO Science Maritimes Region for giving me this golden opportunity that every student will dream of.

Abstract

Operating in a more constrained and more demanding system, DFO has been required to revise and revitalize their strategic plan that brings the requirement of a more transparent, participatory, and informed scientific decision making process through sound science. As for scientific organizations as DFO science, Analytic Hierarchy Process that provides a multiple criteria scoring result based on participants' knowledge and expertise is considered for the science prioritization supporting the decision making. For the specific situation in DFO, considering the organizational arrangement and the delivery of science function, different methods have been formulated and simulated to prove that AHP is an appropriate approach. Meanwhile, to perform the risk analysis for DFO science prioritization, expected utility theory has been chosen as the proper approach to take risk and uncertainty into account. Application software will be developed using these approaches specifically for DFO science risk-based prioritization process and an experiment carried out in Maritimes BIO workshop will be used to demonstrate the approaches and application system proposed in this thesis.

Table of Contents

List of Figures

List of Tables

1. Introduction

This thesis presents an approach for science prioritization specifically for the Department of Fisheries and Oceans Canada (DFO), including methods and software for model construction and system implementation and analysis for supporting science allocation decisions. DFO deals primarily with marine issue including fish and management of the fishery, so the mandate of DFO Science is rather narrow and will be defined accordingly. Therefore, the term "science prioritization" is used especially related to "Science" within DFO.

1.1 Motivation

No business or administrative system succeeds without the ability to identify strategic objectives, prioritize them and then execute them. The problem is, the larger the organization, the more complex the consideration, the more elusive the right choices become. This leads to a costly process that may produce insecure decision making.

When analyzing real life systems, motivated by the need to predict or to control, the decision maker usually faces a complex organizational system of interrelated components, such as various resources, conflictive outcomes or objectives, participants, and devices. Very often, the larger organizational system is broken up into subsystems because the number of entities and their mutual relations increase beyond the ability to manage all the demands placed on the organization. These subsystems, according to their properties, contribute a system with stratified levels, each consisting of many elements, or factors, which is commonly arranged into an organizational hierarchy. The top level of the hierarchy is usually identified as the primary goal of the organization with the lowest level containing the alternatives that toward the goal. The central question in term of the hierarchy is: how strongly do the individual components of the lowest level influence the top factor, the overall goal? Here the determination of the intensities of factors leads to a sequence of prioritization problem within the overall allocation considerations of the organization.

The practice of science prioritization is concerned with weighting alternatives, which fulfill a set of desired objectives. The purpose is to choose the alternatives which most strongly fulfill the entire set of objectives. As well, science prioritization must be carried out subject to a limited set of resources for budgets, and human capital ultimately, decisions on what science projects to undertake proceeds from a clear statement of the strategic goal of the science organization.

In Canada, challenges of national science prioritization, which include limited budgeting, requirements by competing science programs and the different perspectives within different organizational levels, characterize the decisions that must be made by the Department of Fisheries and Oceans (DFO). Since the collapse of groundfish in the 1990s, international tensions, growing recognition of Aboriginal and treaty rights, and unprecedented expansion of the user base of oceans over the past decade, DFO is required to operate in a more constrained and more demanding system. To achieve these outcomes, DFO has been required to revise and revitalize their strategic plan. They have developed new policies and approaches to promote sustainable development and the scientific decision making process, which is directed to become more transparent, participatory, and informed through sound science.

As a federal department, DFO science delivers its mandate and services from national headquarters in Ottawa and its six regional offices: Newfoundland Region, Maritimes Region, Gulf Region, Laurentian Region, Central and Arctic Region, and Pacific Region. A chain of responsibility links the autonomous regions to their own divisions, and within each region, the divisions deliver and are in turn responsible for the delivery of their own science programs, which comprise specific science projects. The hierarchical structure of DFO science can be described as a "top-down" procedure from national to regional to divisional to programs and finally to project levels carried out by individual scientists.

Science prioritization at the different levels proceeds from the strategic goals of the science organization. The ultimate goal of the science prioritization exercise is to develop a procedure for attaching "importance" to the science functions at each level of the hierarchy. Consequently, this importance weighting system will empower the relative resource allocation including funding down to the level of projects throughout the organization.

1.2 Objectives of Research

This research will model the science prioritization process for the case of DFO. Comparative analysis with existing budget and allocation practices will determinate the efficiency of current decisions. The results of this work will provide alternative, effective allocation of scientific resources to achieve the strategic goals of DFO. The purpose of this thesis is to improve the current approach for DFO science resource allocation. To achieve this objective, the research will focus on:

- ❖ Providing better understanding, and a structured approach to problem solving
- ❖ Choosing a measurement method that contributes to associating risk analysis in science prioritization practices
- ❖ Designing hierarchy models appropriate for DFO planning and allocation practices
- ❖ Developing an application system to implement multicriteria priority setting processes for decision making

The allocation of resources including the funding within the hierarchy of DFO will be different according to different points of view concerning the organizational arrangement. In this science prioritization research, alternative science prioritization models that depend on different definitions of the system hierarchy will be analyzed. These models vary in terms of their structure, their functions, or their objectives in the design of DFO science delivery from the perspective of participating science groups.

1.3 Plan of the Thesis

This thesis is composed of the following parts: first in Chapter 1, a general introduction of the science prioritization problem and the specific situation of DFO science prioritization were discussed. Next, the process and methods with examples of the science priority setting exercise and risk analysis from the literature are listed and annotated in Chapter 2. Third, within Chapter 3, different analyses with respect to the way DFO currently defines the risk-based science prioritization problem are introduced and the application of these methodologies to alternative prioritization models is demonstrated. In Chapter 4, modeling and analysis of alternative formulations of the DFO science prioritization are applied using specific software tools developed in this research. The expected results of prioritization models are evaluated in an experiment described in Chapter 5. Finally conclusions and recommendations for future work are found in the contents of the last Chapter in this thesis.

2. Literature Review

The contents of this chapter present historical developments in the approaches dealing with science prioritization. In the section 2.1, an introduction to the general process of solving science prioritization problem demonstrates a sound background for the DFO science prioritization application. Further details concerning methods for multiple criteria science prioritization are discussed in section 2.2. This is followed by a discussion on risk analysis in science prioritization in section 2.3. Some examples from the literature illustrate the past approaches on priorities setting exercise in section 2.4. After discussing the successes and failures of the historical approaches, in the last section, specific considerations for the current research integrating past work with the proposed research are discussed.

2.1 Introduction to Science Prioritization

Priority setting is a process and an important part of strategic planning, a disciplined effort to produce fundamental decisions and actions that shape and guide the organization's direction in response to a changing environment with a focus on the future. It is necessary for an organization to plan its future because research planning always confronts a scarcity of human and financial resources. In scientific organizations, priority setting determines the research portfolio in line with the mission of the organization. Priority setting has to be adapted to the plan it supports, which means that the mandate, the scope and the approximate size of the research effort should be defined before setting the science priorities. Another particularly important aspect of priority setting is the design of a measurement method associated with the context of the priority setting process.

The first consideration in the process of priority setting is its links with planning. The design and implementation of priorities setting will differ depending on whether they are conducted at the national, regional, institute, or program level. A national priority setting exercise is part of planning for long term general research systems development whereas priority setting for an institute is part of that organization's more specific strategic or corporate planning exercise. Secondly, since a participatory approach to the process leads to better decisions and more successful implementation, the principal parties of the organization should be represented in the priority setting process. The last concern is the need for investment in the priority setting exercise, which is always considered in the form of time and money.

Although the specific steps may be undertaken by different levels and types of research, priority setting generally follows four steps (ISNAR 2001):

1. Team building and deciding on participation
2. Defining research objectives, research alternatives and priority setting criteria
3. Measurement and sensitivity analysis
4. Validation and preparing for implementation.

2.2 Methods for Science Prioritization

Science prioritization assists in ranking alternatives through deriving numerical weights for programs or projects with respect to the higher order strategic priorities. Consequently, these weights become meaningful for allocating resources including budgets.

The choice of method is critical to the successful conclusion of a priority prioritization exercise. Research alternatives can be compared by various methods and not all of them are suited for all situations. These methods vary in complexity, flexibility, rigor and other characteristics and can be classified into single criterion methods and multiple criteria methods. The latter methodology is the focus of this thesis research.

2.2.1 Single Criterion Methods

In single criterion methods, the research alternatives are compared by measuring the contribution of each option with respect to a single indicator. There are three commonly used single criterion methods for priority setting (ISNAR 2001):

1. Congruence: the way ranks research alternatives according to the value of a single measure, e.g. the values of production.
2. Benefit/cost analysis: expected research benefits and costs over time are estimated and compared for each alternative.
3. Economic surplus analysis: a refinement of benefit/cost analysis that takes into account market conditions when estimating the benefits.

2.2.2 Multiple Criteria Methods

Science prioritization can also be formulated as constrained multi-criteria decision making problems where decisions must be taken using scare resources to respond to different and competing program needs to reach specific goals. Usually constraints limiting the selection of program possibilities include budget restrictions, the timing and delivery of product, etc. Multicriteria methods compare research alternatives based on various weighted criteria. There are three different groups of multicriteria methods presented below (ISNAR 2001).

2.2.2.1 Scoring Methods

In Scoring Methods, alternatives are compared on several criteria, often using simple scores (e.g., between 1 and 10) but allowing more sophisticated measurement. The importance of the criteria may be differentiated by attaching a weight to each of them.

For example, a specific DFO science program may pursue the following three objectives:

- Sustainable fisheries and aquaculture (SFA)
- Healthy and productive aquatic ecosystems (HaPAE)
- Safe and accessible waterways (SAW)

Research alternatives now need to be compared on these three objectives. In order to do so, first the measurement data is collected following steps:

1. Identify measurable criteria for each of the objectives
2. Identify the relative importance of the different criteria through group

discussion

3. Measure the expected contribution of the different projects to the measurable criteria (assuming that there are three research projects for this program).

For illustration, suppose the results of data collection are as shown in Table 2-1 (adapted from ISNAR 2001).

Table 2-1: Measurement for DFO sample program in Scoring Method

Objective	Criterion	Unit	Weight	Project1	Project2	Project3	Total
SFA	Production gain Processing gain	Million$ Million$	.25 .15	80.0 20.3	7.5 5.1	25.2 7.8	112.7 33.2
HaPAE	Biomass growth Genetic diversity	1000 t 0-9	.30 .10	20.000 6	150.000 0	5.000 2	175.000 8
SAW	Commercial fisheries	Number	.20	3	6	1	10

Secondly, the values of the measurement are converted to percentages to express relative scores or, the share of the different projects in the total contribution to the criterion, which is always 100.0% as shown in the last column of Table 2-2.

Table 2-2: Final score for DFO sample program in Scoring Method

Objective	Criterion	Unit	Weight	Relative score Project1	Relative score Project2	Relative score Project3	Sum of scores
SFA	Production gain Processing gain	Million$ Million$	.25 .15	71.0 61.1	6.6 15.4	22.4 23.5	100.0 100.0
HaPAE	Biomass growth Genetic diversity	1000 t 0-9	.30 .10	11.4 75.0	85.7 0.0	2.9 25.0	100.0 100.0
SAW	Commercial fisheries	Number	.20	30.0	60.0	10.0	100.0
Score				**43.8**	**41.7**	**14.5**	**100.0**

By using percent shares, the scores for all criteria are standardized, so that they can now be multiplied by the criteria weights, which are given as fractions of 1.00

(the sum of all weighting coefficients must always be 1.00). In mathematical terms the final score for each project can now be expressed as:

$$Y_i = \sum_{j=1}^{n} w_j \times y_{ij} \tag{1}$$

where Y_i = final score of research alternative i

w_j = weight of criterion j

y_{ij} = the relative score of research alternative i on criterion j

n = the number of criteria

For example, the final score for Project 1 is calculated as:

$$Y_1 = 0.25\times71.0 + 0.15\times61.6 + 0.30\times11.4 + 0.10\times75.0 + 0.20\times30.0 = 43.835 \tag{2}$$

Eventually, the final score for projects is calculated as shown in Table 2-2. As for the results of the calculation, Project 1 is favored and Project 2 is only slightly less favored which is still much higher than Project 3. Sensitivity analysis may now help to evaluate the impact of different weighting systems by changing weights or parameters (e.g., prices).

2.2.2.2 Analytic Hierarchy Process

Analytic hierarchy process (AHP) is a mathematically more rigorous version of scoring methods (Saaty 1988). It is a comprehensive, logical and structural framework that improves the complex decisions following three steps (ISNAR 2001):

(1) Decompose the problem in a hierarchical structure incorporating all relevant decision criteria. The basic structure of hierarchy is shown in Figure 2-1. The goal of the decision, such as "Science Prioritization", is at the top of the hierarchy. The level next to the goal consists of the criteria relevant for it, and at the bottom level are the alternatives such as research projects to be evaluated.

(2) Make comparative judgment of the elements in the hierarchy by pairwise comparison, which explicitly recognizes and incorporates the knowledge and expertise of the participants. The alternatives and the criteria are compared in pairs with respect to each element of the next higher level. The fundamental scale

listed in Table 2-3 can be used to express the comparisons in verbal terms which are then translated in the corresponding numbers.

Figure 2-1: The basic structure of the hierarchy of AHP model

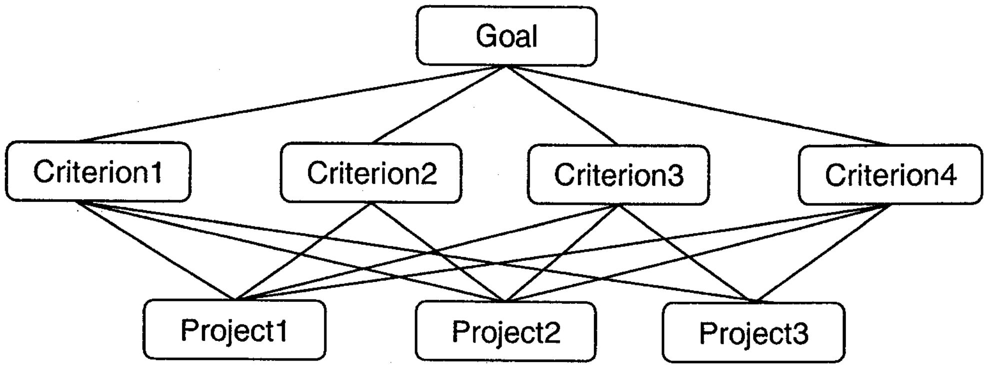


Table 2-3: Fundamental scale for pairwise comparisons of AHP model

Verbal scale	**Numerical values**
Equally important, likely or preferred	1
Moderately more important, likely or preferred	3
Strongly more important, likely or preferred	5
Very strongly more important, likely or preferred	7
Extremely more important, likely or preferred	9
Intermediate values to reflect compromise	2, 4, 6, 8

(3) Synthesize the comparisons to obtain the priorities, integrating subjective judgments but also objectively measured information where it is available. The overall priorities of the alternatives are calculated with respect to each criterion and the weights of each criterion with respect to the goal.

The crux of AHP method is the determination of the relative weights to rank the decision alternatives. Assuming that we are dealing with n elements of a level in a given hierarchy, we need to find their weights of influence, $w_1, w_2, w_3, \cdots, w_n$, on elements in the next level. In the ideal case, we can express their relative weights in a $n \times n$ pairwise comparison matrix, A:

$$A = \lfloor a_{ij} \rfloor, \quad i,j = 1, 2, \cdots, n \tag{3}$$

and

$$a_{ij} = \frac{w_i}{w_j}, \quad i,j = 1, 2, \cdots, n \tag{4}$$

then

$$A = \begin{bmatrix} w_1/w_1 & w_1/w_2 & w_1/w_3 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_3 & \cdots & w_2/w_n \\ w_3/w_1 & w_3/w_2 & w_3/w_3 & \cdots & w_3/w_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & \cdots & w_n/w_n \end{bmatrix} \tag{5}$$

and by taking the matrix product of the matrix A with the vector w , we have

$$\begin{bmatrix} w_1/w_1 & w_1/w_2 & w_1/w_3 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_3 & \cdots & w_2/w_n \\ w_3/w_1 & w_3/w_2 & w_3/w_3 & \cdots & w_3/w_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & \cdots & w_n/w_n \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} nw_1 \\ nw_2 \\ nw_3 \\ \vdots \\ nw_n \end{bmatrix} = n \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} \tag{6}$$

which is same as $Aw = nw$. The matrix A is called consistent since

$$a_{ik} = a_{ij} \times a_{jk}, \text{ for all } i, j, k \tag{7}$$

Solving for a nonzero solution to this set of equations is well known as an eigenvalue problem. The eigenvector provides the priority ordering, and the eigenvalue is a measure of the consistency of the judgment (Nicholson 1990).

In the general case, a_{ij} will deviate from ideal ratio w_i / w_j and the strong consistency property most likely does not hold. (This refers to small errors and inconsistencies in judgments). It has been known that for any matrix, small perturbations in the entries imply similar perturbations in the eigenvalues, thus the eigenvalue problem for the inconsistent case is:

$$Aw = \lambda_{\max} w \tag{8}$$

where $\lambda_{\max}$ will be close to n (greater or equal) and other λ's will be close to zero. The estimates of the weights for the elements are:

$$w_i = \frac{1}{\lambda_{\max}} \sum_{j=1}^{n} a_{ij} w_j \qquad i = 1, 2, \cdots, n \tag{9}$$

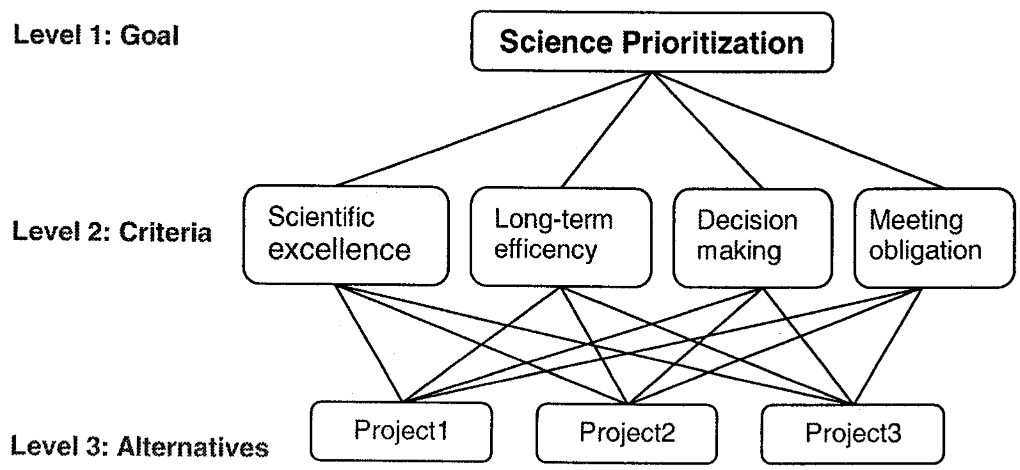
The calculation of weights can be done by normalizing the matrix, which is shown in a later example. Within the process, consistency means that the decision maker is exhibiting coherent judgment in specifying the pairwise comparison of the criteria or alternatives. In the general case, the closer $\lambda_{\max}$ is to n, the more consistent the judgments. Saaty (1988) defined the consistency index:

$$CI = \frac{\lambda_{\max} - n}{n-1} \tag{10}$$

as the indicator of "closeness to consistency". In general, if this number is less than 0.1, the consistency of the judgments may be thought to be satisfactory.

Consider a simplified science prioritization example which has three projects, Project1, Project2 and Project3. The criteria for the analysis of the projects are efficiency, sustainability, strategic effects (indirect research effects), and feasibility. The hierarchy of the problem is structured as shown in Figure 2-2.

Figure 2-2: The hierarchical structure of the sample decision problem used to demonstrate AHP method



Now for the comparisons of the elements in the hierarchy, we can start with the projects, which are compared with respect to each criterion. For instance, the total of comparisons with respect to the criterion "efficiency" generates a matrix as presented in Table 2-4.

Table 2-4: Pairwise comparison for the efficiency criterion in AHP sample decision problem

	Project1	Project2	Project3	Local priority
Project1	1	4	2	0.571
Project2	1/4	1	1/2	0.143
Project3	1/2	2	1	0.286

The local priorities (to three decimal places) can be calculated by normalizing the matrix (the sum of local priorities is equal to 1.00) following the steps:

Step 1: Squaring the matrix:

$$\begin{bmatrix} 1.000 & 4.000 & 2.000 \\ 0.250 & 1.000 & 0.500 \\ 0.500 & 2.000 & 1.000 \end{bmatrix} \times \begin{bmatrix} 1.000 & 4.000 & 2.000 \\ 0.250 & 1.000 & 0.500 \\ 0.500 & 2.000 & 1.000 \end{bmatrix} = \begin{bmatrix} 3.000 & 12.000 & 6.000 \\ 0.750 & 3.000 & 1.500 \\ 1.500 & 6.000 & 3.000 \end{bmatrix} \tag{11}$$

Step 2: Computing the eigenvector

(1) Sum the row of the result matrix in step 1

$$\begin{bmatrix} 3.000 & 12.000 & 6.000 \\ 0.750 & 3.000 & 1.500 \\ 1.500 & 6.000 & 3.000 \end{bmatrix} \Rightarrow \begin{bmatrix} 21.000 \\ 5.250 \\ 10.500 \end{bmatrix} \tag{12}$$

(2) Sum the row totals

$$21.000 + 5.250 + 10.500 = 36.750 \tag{13}$$

(3) Divide the row sum by row totals

$$\begin{bmatrix} 21.000/36.750 \\ 5.250/36.750 \\ 10.500/36.750 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.571 \\ 0.143 \\ 0.286 \end{bmatrix} \tag{14}$$

Step 3: Iterate step 1 and step 2 for the resulted matrix of matrix multiplication until the eigenvector does not change from the previous iteration under the certain tolerance, which in this sample is set to three decimal places.

The final result of these computations is the eigenvector of pairwise comparison matrix in Table 2-4. It gave us the relative ranking of alternative projects. The illustration used here is a simple one that its eigenvectors exist and can be determined by a simple squaring of the target matrix. In fact, depending on the tolerance, it may require 3rd order and 4th order matrix multiplications to get to the eigenvector.

The same procedure is used for the comparisons of the criteria with respect to the overall goal. The matrix with the hypothetical values and the resulting weights are shown in Table 2-5.

Table 2-5: Pairwise comparison of the criteria with respect to the goal in AHP sample decision problem

	Efficiency	**Sustainability**	**Strategic effects**	**Feasibility**	**Weights**
Efficiency	1	5	4	2	0.483
Sustainability	1/5	1	1/3	1/5	0.066
Strategic effects	1/4	3	1	1/3	0.136
Feasibility	1/2	5	3	1	0.315

Table 2-6: Priorities, weights, and the final ranking of the projects in AHP sample decision problem

	Local priorities with respect to the criterion (weights):				
	Efficiency (0.483)	**Sustainability (0.066)**	**Strategic effects (0.136)**	**Feasibility (0.315)**	**Global priority**
Project1	0.571	0.452	0.653	0.085	**0.422**
Project2	0.286	0.072	0.062	0.644	**0.354**
Project3	0.143	0.476	0.285	0.271	**0.224**

Once the comparisons are completed, the global priorities of projects (for all the criteria) are determined by the sum of the multiplication of local priorities of the project with respect to each criterion and the criterion weight with respect to the goal (Table 2-6), e.g., the global priority of Project1 is:

$$0.571 \times 0.483 + 0.452 \times 0.066 + 0.653 \times 0.136 + 0.085 \times 0.315 = 0.422 \qquad (15)$$

Sensitivity analysis now can be performed to evaluate the influence of variables on the weights and resulting ranking.

2.2.2.3 Mathematical Programming Methods

For Mathematical Programming Methods (MAP), the best research alternatives are selected according to predefined criteria weights, under certain restrictions, e.g., on resource availability. MAP aims to maximize an objective function (the contribution of research alternatives to a set of weighted criteria). It considers explicitly the constraints under which the objective function is being maximized. The two main constraints to be considered in MAP are research capacity and policy restrictions, such as the limited operational budget (ISNAR 2001).

The mathematical expression of a MAP problem is expressed as follows (Alston, Norton and Pardey 1995):

Max $Z(x) = G[z_1(x), z_2(x), \cdots z_k(x)]$ (16)

Subject to: $x \in X$, $x \geq 0$,

where

$Z(x)$ = the objective function with k objectives $z_i(x), i = 1, 2, \cdots k$

G = the goal operator defining the objective function being maximized

x = the n-dimensional vector of decision variables, i.e., research programs to which resources are committed to achieve the stated objectives, hence, they are non-negativity.

X = the decision space, i.e., the set of research resources available and any other constraints on the choices of x

It is important to make sure that the same concern is not expressed in both the multiple objective function and in the constraints. Once a certain level of satisfaction has been achieved, there is no further concern.

2.3 Risk Analysis

It is often the case that in a given decision problem, the outcome of decisions depend on occurrences beyond the control of decision maker that may have either desirable or undesirable consequences. These occurrences mean that fundamentally the problem outcome is "uncertain" and that decision makers must therefore exercise their attitude toward "risk" in planning for decisions in such an uncontrollable environment. Since the uncertainty associated with the decision problem is unavoidable, it becomes necessary for decision makers to deal with the potential consequences aligned with a decision and its uncertain outcome, or on the other hand, to put the possibilities of undesirable occurrence into consideration during the decision making process (Bertsekas 1976).

2.3.1 Risk Analysis in General

Decision making problem under uncertainty possesses many important characteristics that are not present in deterministic decision problem, i.e. a problem whose decisions yields outcomes which are not certain. The most prominent characteristics include the need to take into account the level of a decision maker's acceptable risk in the decision problem formulation and the possibility of feedback such as information gathering during the decision making process.

To illustrate the first characteristics, consider an investment problem which needs to allocate one unit of capital between two investment opportunities A and B. Investment A yields \$1.49 per dollar invested with certainty, and opportunity B yields \$0 per dollar invested with probability 0.8 and \$7.5 per dollar invested with probability 0.2. The problem is to make a decision on investment proportions between these two opportunities. If the objective is to simply maximize the average profit, then the whole capital should be invested in B with its calculated expected value, *E* of \$1.5 per dollar invested calculated in the standard way as follows:

$$E = 0.8 \times 0 + 0.2 \times 7.5 = 1.5 \tag{17}$$

where *E* indicates the expected value of the return from making investment B.

However, many investors would not want to risk losing all their capital by investing in B, and some might even invest exclusively in the sure, but lower opportunity A. More generally, investors would prefer to invest a certain fraction in both A and B.

The maximum expected profit formulation for this investment decision problem is then inappropriate because the risk associated with each allocation and borne in the attitude of each decision maker is not reflected at all.

This sample problem indicates that a problem formulation for decision making under uncertainty and risk should consider all possible outcomes associated with a given strategy as well as the inherent attitude of the decision maker with respect to the occurrence of these possible outcomes. Thus, to formulate decision problems under uncertainty properly is to formulate the problem in the form that consists of:

- The set of possible decisions or alternatives
- The set of uncertainties or possible states of the problem
- The set of possible outcomes of the decision problem
- The function that determines which expected outcome will result from a given decision and a given uncertain state.
- The decision maker's relative preference among the possible outcomes

2.3.2 Approaches to Risk Analysis

There are a number of approaches and viewpoints for determining a ranking among decisions including Payoff Functions, Dominant and Non-inferior Decisions, the Min-Max Approach and Expected Utility Approach (Bertsekas 1976). This section deals with the Expected Utility Approach, which is chosen specifically for the risk analysis in DFO science prioritization.

It is usually the situation that in a given decision making problem under uncertainty, the decision makers are in a position to know that the uncertainty occurs in accordance with a given probabilistic mechanism depending on the decision adopted. To be specific, assume the decision problem here is formulated into the decision table form as discusses above:

D The set of countable possible decisions

N The set of uncertain countable states

O The set of outcomes

$f: D \times N \rightarrow O$ The function

$\prec$ The relation, e.g. for $o_1, o_2 \in O$, $o_1 \prec o_2$ means that outcome o_2 is at least as preferable as outcome o_1

Each decision $d \in D$ specifies the probability P_d of each outcome $o \in O$ via the function $f(d,n)$, where $n \in N$. This probabilistic mechanism can be expressed as:

$$P_d(o) = P(\{n \mid f(d,n) = o\} \mid d) \qquad \forall o \in O \tag{18}$$

The basic idea of the expected utility approach is assuming that the decision maker had a complete ranking of the outcomes and the probability measure on the set of outcomes, i.e. the decision maker is capable of expressing his or her preference between any two probability distributions on the set of outcomes, then a complete ranking of all decisions can be obtained by means of the relation

$$d_1 \prec d_2 \quad \Leftrightarrow \quad P_{d_1} \prec P_{d_2} \tag{19}$$

Furthermore, the ordering of decisions can be expressed by means of an essentially unique numerical function called the utility function U, which maps the space of outcomes into the set of real numbers and satisfies

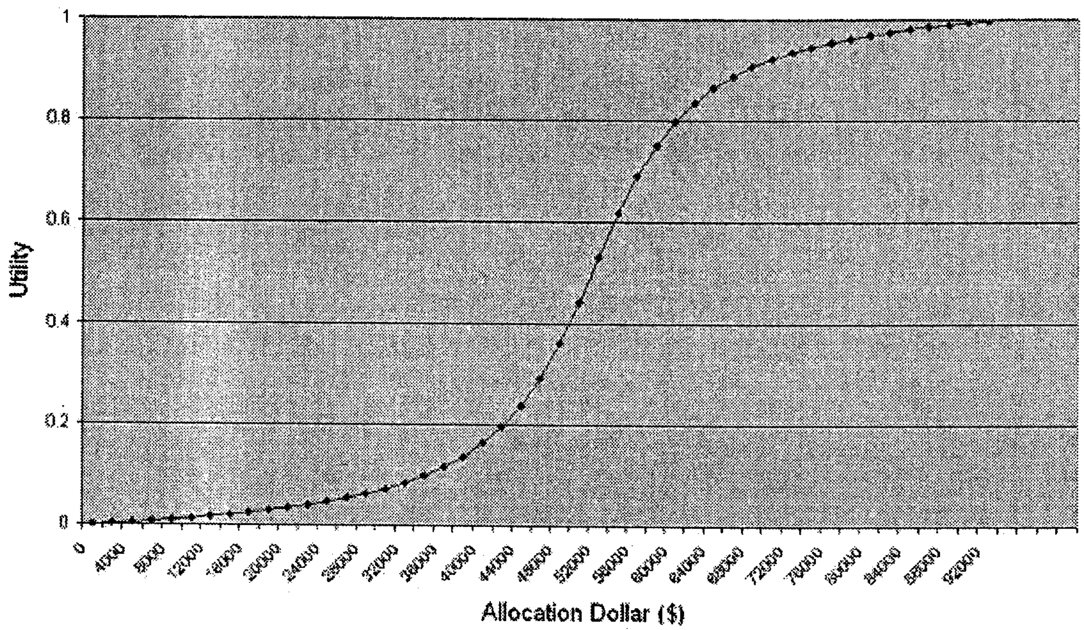
$$d_1 \prec d_2 \quad \Leftrightarrow \quad P_{d_1} \prec P_{d_2} \quad \Leftrightarrow \quad E\{U[f(d_1,n)] \mid d_1\} \le E\{U[f(d_2,n)] \mid d_2\} \tag{20}$$

where the expectations are taken with respect to the corresponding probability distribution P_d on N. It is proved in the expected utility theorem that a preference relation on the set of probabilities mechanism satisfying certain assumptions can be characterized numerically by means of the utility function.

Different decision makers participating in the same decision problem may have different utility functions, so that the form of the utility function must be specified before the problem can be solved numerically. This utility function can usually be found experimentally if necessary.

Consider a decision maker possessing an s-shaped utility curve (Figure 2-3), generally it is denoted that the decision maker is risk averse which is shown equivalent to concavity of the utility function, and is risk preferring, which is in the case of a convex utility function. The attitude of risk aversion characterizes most decision makers in normal situations. A gambler playing an unbiased roulette and receiving no expected reward is a typical example of a risk preferring decision maker. A decision maker having a linear utility function is said to be risk neutral.

Figure 2-3: S-shaped utility curve sample for decision maker



The notion of risk captures a basic attribute of the attitudes of the decision maker and often characterizes important aspects of his behavior. Pratt (1964) proposed a widely accepted measure of risk as follow:

$$r(x) = -U''(x)/U'(x) \qquad (21)$$

where $U''(x), U'(x)$ denote the second and first derivative of U, respectively, and assuming that $U' \neq 0$. This function measures locally (at the point x) the risk aversion of the decision maker and is called the index of absolute risk aversion. It is generally accepted that $r(x)$ is a decreasing or at least nonincreasing function of x, i.e. an individual is risk preferring as his wealth is increased.

2.4 Past Approaches with Applications

This section provides examples of science prioritization or other prioritization exercises that contain descriptions of methodology, process and results. Examples are taken from the United States Space program of National Aeronautics and Space Administration (NASA), the US climate change science program, the U.S. Department of Veterans Affairs decision process and the U.S. Northeast Fisheries Science Center (NEFSC) resource allocation.

2.4.1 NASA Space Program Priorities

NASA has a stake in some of the biggest intellectual problems in science: the origin of life, the nature of the solar system, human exploration outside the planet, and the characterization of Earth from space. In several areas of biological and physical research, solutions of very large, important questions require microgravity. To assess NASA's plans to maximize the microgravity research conducted by the agency's Office of Biological and Physical Research (OBPR), NASA administrators have selected an independent task force, the Research Maximization and Prioritization Task Force (ReMAP), to prioritize among research themes (NASA 2002).

The ReMAP committee is comprised of distinguished multidisciplinary experts who provide extensive background information and offer constructive comments and suggestions. It serves as a follow on to the International Space Station (ISS) Management and Cost Evaluation (IMCE) Task Force that evaluates the budget and management challenges facing the ISS Program. Part of the IMCE charter was to advise NASA and the Administration on how to maximize the scientific returns on station while operating within the guidelines of the President's Fiscal 2002 Budget Blueprint.

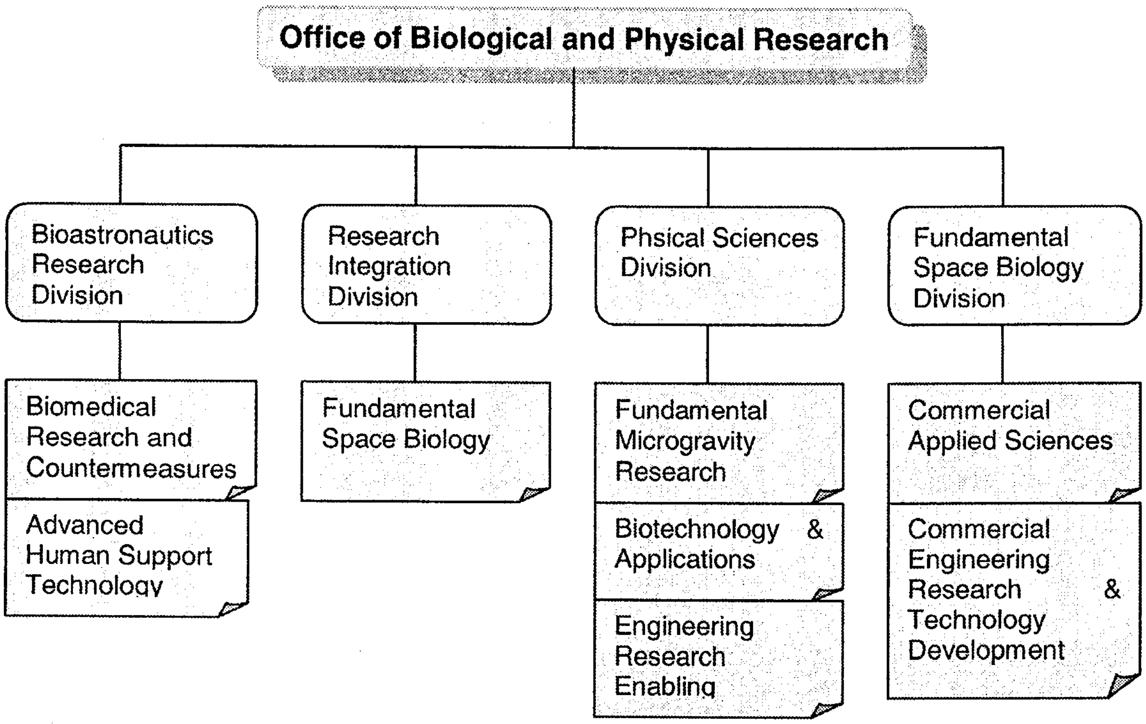
NASA presented the research of the 4 Divisions of the Office of Biological and Physical Research (OBPR) to the Task Force as comprising 8 primary theme areas, integrated organizationally shown in Figure 2-4.

The OBPR program background, objectives, and descriptions of the research thrusts and theme areas were provided to the Task Force. OBPR first developed a proposed set of criteria for evaluating OBPR research programs:

- Scientific Importance
- Impact on Scientific and Technological Community
- Relevance to a Broad Constituency

- Contributions to National Goals

Figure 2-4: OBPR research themes in NASA space program



These criteria were vetted by the Task Force and were agreed to be an appropriate starting point for evaluation of research priorities. The Task Force incorporated all of the above into their prioritization process, but went a level further in deciding the final priorities. Additional considerations for distinguishing between high, medium, and lower priorities were articulated by the Task Force. Each Disciplinary Team established priorities and presented their rationale for prioritization to the full Task Force for discussion. This included presentations by individuals or small groups within the Team summarizing the Task Force perspective on previous reviews and recommendations ("meta-analysis") of NASA's space research program in the theme area. Once within-theme priorities had been identified, the Task Force, as a body, analyzed the results across theme areas and developed a set of science-driven priorities for OBPR research.

The ReMAP committee for the first time establishes priorities and goals for OBPR and ISS research across disciplines. The findings and recommendations in the ReMAP final report provide a framework for prioritizing a productive research program for OBPR and ISS. It is noteworthy that the committee was successful, during meeting deliberations, in establishing a rationale and strategies for prioritization of the overall research program for OBPR and for ISS (NASA 2002).

2.4.2 Climate Change Priorities

In February 2002, President George W. Bush announced the formation of a new management structure, the Climate Change Science Program (CCSP), to coordinate and direct the US research efforts in the areas of climate and global change. The CCSP aims to balance the near-term (2- to 4-year) focus of the Climate Change Research Initiative (CCRI) with the breadth of the US Global Change Research Program (USGCRP), pursuing accelerated development of answers to the scientific aspects of key climate policy issues while continuing to seek advances in the knowledge of the physical, biological and chemical processes that influence the Earth system (CCSP 2002).

Even with substantial budgets, the potential scope of climate change research, observations, and scientific synthesis is so large that the CCSP must clearly identify the highest priority activities for support. Therefore, prioritization is an ongoing process in the program and reflects changes in needs and scientific progress and opportunity. Initial CCSP priorities have developed in response to the NRC report, Climate Change Science: An Analysis of Some Key Questions, characterized areas of uncertainty in scientific knowledge concerning climate change, and identified research areas that will advance the understanding of climate change. Based on this NRC report, the administration established the CCRI to study areas of uncertainty about global climate change and identify priority areas where investments can make a difference.

In developing priorities, the CCSP has considered information from many sources, including:

- The analysis and reporting requirements imposed by the Global Change Research Act of 1990
- The focus on reducing key scientific uncertainties through the Climate Change Research Initiative of June 2001

- The recommendations in various reports of the National Research Council
- Climate change assessment reports by the Intergovernmental Panel on Climate Change
- The annual program plans of the 13 collaborating CCSP agencies and departments, as documented in the annual series of *Our Changing Planet* reports
- The *Discussion Draft Strategic Plan* published by the CCSP in November 2002
- The deliberations at the Climate Change Science Workshop in December 2002, which was sponsored by the CCSP and attended by 1,300 climate specialists
- The written comments submitted after the December 2002 CCSP workshop.

The CCSP employ the following overall criteria in establishing priorities for work elements selected for support:

- Scientific or technical quality
- Relevance to reducing uncertainties and improving decision support tools in priority areas
- Track record of consistently good past performance and identified metrics for evaluating future progress
- Cost and value

The research priorities of the CCSP are reviewed on an annual cycle through the budget process and reflect priority needs and scientific opportunities. While the CCSP Strategic Plan includes a decade-long strategy, it also establishes priorities for the near term consistent with the CCRI. These priorities have been established in response to information from the sources listed above.

Furthermore, the CCSP plan calls for the creation of a series of synthesis and assessment reports. These reports represent principal responses to the top-priority research, observation, and decision support needs described above. In addition to the planned synthesis and assessment reports, the CCSP collaborating agencies continue to sponsor a large number of research projects each year. Prioritization principles also are applied to the entire group of projects in each of the CCSP study areas (CCSP 2002).

2.4.3 VA Decision Process

As with most other agencies in the US federal government, the Department of Veterans affairs (VA) is an organization with a variety of officers annually competing for scarce resources. Prior to 1997, the Veteran's Administration (VA) largely developed its project priorities manually in table discussions with little structure. Decision makers considered strategic objectives but established no explicit link between strategic priorities and funding requests. The manual process resulted in a priority list that lacked an audit trail describing how projects were ranked (Wasyluk and Saaty 2003).

In 1997, executives at VA set out to develop a completely integrated capital investment process in order to meet several legislative mandates to improve their decision making process for capital in vestment Legislation such as the Government Performance Results Act (GPRA) of 1993, the Federal Acquisition Streamlining Act (FASA), etc. VA decision makers have chosen an off-the-shelf prioritization tool based on the Analytic Hierarchy Process (AHP) method of decision making. Over the next three years, the VA designed its organization to 1) review a variety of investment options from construction projects to information technology projects and 2) standardize the application for capital investments and develop a capital investment methodology comprised of formulation and execution reviews.

Formulation review is comprised of three phases: functional development, technical review, and strategic review. The functional development phase is at the field level, where needs are realized, gap analysis completed, application proposals developed, and solutions ultimately applied. In the technical review phase, proposals receive early rounds of technical and financial scrutiny from department-wide councils or boards, as well as some initial prioritization within the owner organization. Finally, the strategic review phase includes a two step review process and prioritization by the VA Capital Investment Board (CIB).

The Execution Review includes the confirmation phase and occurs once funds are appropriated and included in the president's budget. Planning assumptions for these funded proposals were made 18-24 months earlier. For example, prior to the obligation of funds, the VA Capital Investment Board (VACIB) will review the upcoming fiscal year's obligations to assure that the proposed investment is still needed and that the planning assumptions used are still valid VACIB prioritizes the projects into a single list according to how they contribute to strategic objectives.

The process resulted in a dramatic reduction in decision making time, better proposals, and overall better selection and control of projects funded based on their contribution to strategic objectives. The success of the methodology was due largely to the organization's commitment to improve the way it establishes the gap analysis and plans its projects, and the VA expects that the process will continue to improve over time due to the auditable framework they have now implemented for capital investment decisions (Wasyluk and Saaty 2003) .

2.4.4 NEFSC Resource Allocation

The Northeast Fisheries Science Center (NEFSC) is the research arm of the National Oceanic and Atmospheric Administration (NOAA) Fisheries in the United States. The center plans, develops and manages a multidisciplinary program of basic and applied research to better understand living marine resources of the Northeast Continental Shelf Ecosystem from the Gulf of Maine to Cape Hatteras, and the habitat quality essential for their existence and continued productivity. It describes and provides to management, industry and the public, options for the conservation and utilization of living marine resources, and for the restoration and maintenance of marine environmental quality (Expert Choice 2000a).

The goal of the science prioritization process for the NEFSC is to provide advice on how to apply most effectively the Center's resources (vessels, personnel, funds, and equipment) to research operations. An AHP model was developed to assist in planning and evaluating the Center's research projects, and advice on science operations in precisely defining critical information needs on how to rank the priorities and measure the performance of each project. The resulting allocation plan directly influence the contribution of individual science projects to meet the NOAA Fisheries strategic goals.

The hierarchical structure of NEFSC resource allocation problem is shown in Figure 2-5 using the results from the AHP and the software package "Expert Choice". The long-term strategic goals (objectives) of NEFSC science research are defined in the second level of the hierarchy as:

- Rebuild overfished and maintain productive fisheries
- Advance fishery forecasts and ecosystem models
- Integrate management of protected species and fisheries
- Improve seafood safety
- Improve fishery habitat conservation
- Management needs to operate NEFSC

Under each objective, multiple criteria are listed for the assessment of final ranking in the third level of hierarchy. Alternative projects listed below are prioritized in this resource program according to their contribution to the objectives and the ultimate goal:

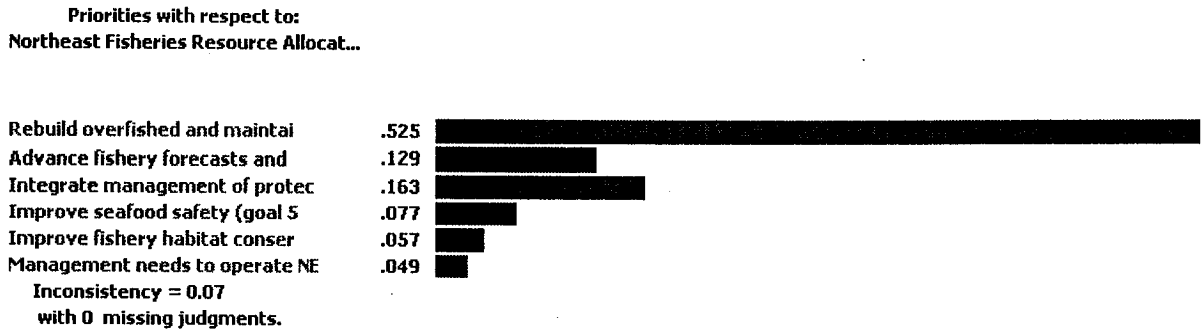
— PopDyL-1
— Calibrate
— Sampling Dat
— Rec Data
— Branch Needs
— Eval Effort
— Fishbio 1
— Survey 1
— DMS 1
— Multisp Mode

The result of the AHP program is a set of priorities ranking according to the goal, objectives, and criteria (e.g. Priorities with respect to the goal shown in Figure 2-6) from Expert Choice 2000. Based on this priorities ranking, the synthesis report aligns the resources behind the most effective means to achieve the competing objectives considered in this program.

Figure 2-5: Resource allocation hierarchy of NEFSC sample model

- Northeast Fisheries Resource Allocation
  - Rebuild overfished and maintain productive fisheries
    - Fisheries data from the commercial and recreational industry
    - Biological data from NEFC research vessels (age&growth,fecundity)
    - Stock assessments (including management advice)
    - Economic assessments
  - Advance fishery forecasts and ecosystem models
    - Enviornmental models (biotic and abiotic mechanisms)
    - Pollution fate and effects models
    - Fishery forecasting model (technical interactions, etc.)
    - Bio-economic models
    - Fisheries biology
  - Integrate management of protected species and fisheries
    - Population assessments (OSP, etc.)
    - Human interactions (fishing, incidental take, trophic effects)
    - Ecological roles/habitats (food habits, spatial distrib. etc...)
  - Improve seafood safety
    - Fish and invertebrate biohazards (algal blooms, PSPs, etc.)
    - Hazardous toxic chemicals (PCBs, metals, hydrocarbons)
    - Quality of the seafood (taste and odor problems,nutritonal value)
  - Improve fishery habitat conservation
    - 12 and 106 mile studies
    - Functional value of habitat to LMRs
    - Contaminant effects on population (organic and inorganic)
    - Nutrient eutrophication (Low dissolved oxygen)
  - Management needs to operate NEFC
    - Resource needs (fiscal/personnel) to administer NEFC research
    - Plan development/implementation of NEFC current & future research
    - Information services (library, aquarium, public relations)
    - Facilities, equipment, supplies, maintenance
    - Computer support (hardware, software, consultation, contracts)

Figure 2-6: Priorities with respect to the goal of NEFSC sample model



2.5 Consideration for Current Research

The strength of multiple criteria methods is that they incorporate many different concerns and make most decisions not based on the assessment of one single criterion. By allowing different criteria and by jointly defining weights, a priority setting working group may increase its understanding of the issues and how they interact, and may improve the degree of consensus. Since multiple criteria methods use a range of criteria, they are normally better suited to compare research alternatives with widely different characteristics.

Scoring methods are easy to apply and understand. They can combine subjective judgments with objective assessments and will not be halted due to the absence of data. By basing the final judgment on a combination of scores, the possible mistakes on each criteria are somewhat compensated. On the other hand, they often neglect the linkages between criteria and the actual research objectives, and the linkages among the criteria and attributes. Consequently the identification of the weights is passed over very quickly, or is completely forgotten.

Mathematical programming methods are direct optimization methods. They represent in principle the most advanced methodology, because other approaches such as multiple criteria and economic surplus methods may be subsumed in a MAP framework. Nevertheless they are not often used. The mathematical complexity including the difficulty to justify nonlinearities makes it inaccessible for many problems. The method is resource-intensive and it is often difficult to set realistic constraints and to interpret the optimization results.

AHP has a major impact on the understanding by the participants of the factors which influence the value of a project. Besides a broad agreement on the ranking of projects, the process provides profound insight in the art of complex decision making and encourages the participants to pool their knowledge and expertise. Based on past successes and comparable applications, AHP is judged to be more suitable for the situation in the DFO prioritization process. Although in AHP prioritization, the pairwise comparison is not based on exact measurement, but also on subjective judgments, the synthesis results then may have inconsistencies among judgments. In general we will still be satisfied with the result if judgments may be considered "consistent", which means that the small errors and inconsistencies across participants are thought of as acceptable in subjective judgments.

As for the samples discussed in the previous section, the resource allocation program of Northeast Fisheries Center is a successful application using the structured hierarchy (AHP) to analyze the problem. The situation of NEFSC resource allocation is parallel with the DFO science prioritization problem, which is also aiming at the effective process for resource allocation down to the level of individual projects. From the success of this application, we can see that the AHP multiple criteria model is a preferred methodology for the science prioritization of DFO. The following research will be developed based on this methodology specifically for DFO case.

3. Methods for DFO Science Priority Setting

The purpose of this chapter is to incorporate the methodology introduced above in the DFO risk-based science prioritization problem. The situation of DFO science prioritization is introduced as the background information in the first section. Then based on the understanding of this situation, AHP models of different levels are designed to fit the strategic perspectives of DFO in section 3.2. Furthermore, three detailed model formulations are illustrated and risk is specified for DFO situation in the following sections respectively. Finally, these methodologies and problem formulations are applied into DFO science risk-based priorities setting problem to formulate a process especially for current situation.

3.1 Strategic Planning at DFO

Canada's fisheries, oceans, lakes, and waterways form an important economic engine that contributes to the livelihoods of Canadian communities throughout the country, and to Canada's economy as a whole. Fisheries and Oceans Canada, on behalf of the Government of Canada, is responsible for policies and programs in support of Canada's economic, ecological, and scientific interests in the oceans and freshwater fish habitat; for the conservation and sustainable utilization of Canada's fisheries resources in marine and inland waters; and for safe, effective, and environmentally sound marine services responsive to the needs of Canadians in a global economy (Canada 2003).

Fisheries and Oceans Canada has seen considerable change over the past decade. There has been growing recognition of environmental challenges such as pollution, species at risk, and climate change. Facing the challenges of insufficient resources to address current mandate and increasing demands, increasing costs of science and requirements for integrated management approaches, DFO realizes that there is a need to change the way it does business.

The Department is continually re-evaluating its mandate and key directions (DFO 2003c). Ensuring safe, healthy, productive waters and aquatic ecosystems for the benefit of present and future generations is essential to the department's activities, which are constructed aiming at these strategic objectives:

- Sustainable fisheries and aquaculture
- Healthy and productive ecosystems
- Safe and accessible waterways

To achieve these objectives, science prioritization that uses a systematic approach in assessing risk and provides a better rationale in resource alignment is necessary. The risk areas explicitly identified on "Risk-based Approach to Priority Setting" (DFO 2003b) also presented at the Oceanstone Maritimes Region Science Retreat by HQ Science Policy included the three strategic objectives referred to above, along with 4 criteria as follows:

1. Scientific excellence and credibility
2. Long-term efficiency
3. Sound and relevant decision-making
4. Meeting obligations

Further elaboration of these "risk areas" was provided in the document "Model Overview – Risk based Priority Setting Model" (DFO 2003e). This document was developed with the primary purpose of describing the priority setting tool to "help national priority setting for Science Programs and Activities at the Science Sector level". It states that the tool "uses 7 risk areas that are common to all Science programs and functions to determine the type of risks." The common risk areas of interest are defined more explicitly in the document as follows:

1. <u>Scientific excellence and credibility or Measuring scientific excellence and credibility</u> – excellence is crucial in maintaining and restoring confidence in science program's capacity. It concerns the knowledge foundation as well as the soundness and accuracy of information that are necessary for science advice and information.

2. <u>Long-term efficiency or Producing long-term efficiency</u> – concerns how a program/project the measures taken now in perspective of a long-term gain in efficiency. This includes the use of new technology and efforts to achieve

partnering in the right places. While these measures have immediate cost implications, the investment is made with the hope that there will be advantages, in time, in terms of quality and cost.

3. Sound and relevant decision-making or Making sound and relevant decisions – links to provision of sound, relevant and timely scientific advice and information for decision-making of the Department

4. Meeting obligations or Fulfilling obligations/mandate – related to the degree to which a program/project contributes to fulfilling obligations stipulated in various Acts and Regulations as well as meeting commitments, policies, government and departmental priorities, international, regional and national obligations

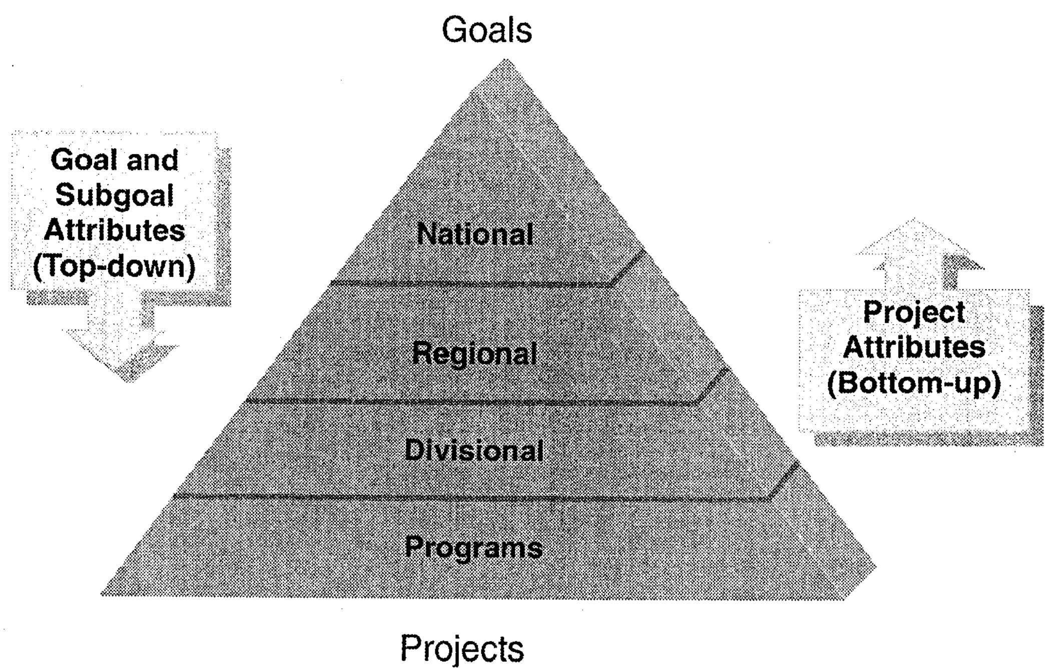
In recent decades, the "systems approach" to problems in the social and behavioral sciences has found its place next to the older reductionism methods which seem more appropriate to the physical sciences. The system can be defined in terms of the interactions of the components. According to the functions of the parts, the objectives they are intended to fulfill, the higher objectives they are leading up to, decide what objectives are being satisfied and what conflicts among individuals may have to be resolved. An alternative view to understanding the system is based on its organizational structure, according to the physical arrangement of the parts and the flow of material and personnel which define the relations and dynamics of the structure. These are the original ideas applied to systems analysis methods (Klir 1994, Wymore 1993).

For the situation in DFO, there are several models for science prioritization based on a systems understanding including institutional problem formulation, functional problem formulation, and multidisciplinary problem formulation. The institutional view assumes that allocation decisions are based on the structure of the organization, and the regions could be considered to be more or less autonomous and may even exert influence to provide different expertise or service. The functional view assumes that the national perspective actually provides specific process direction to be incorporated at each region where the regions are capable of delivering identical and coordinated services. Finally, the multidisciplinary view defines the whole system as one integrated hierarchy, instead of the level model talked above. These perspectives imply different AHP level models which will be illustrated in the later sections.

3.2 AHP Model of DFO Science Prioritization

Consider an organizational arrangement illustrating the delivery of DFO science proceeding in a hierarchy from National to Regional to Divisional to Programs and ultimately to Projects carried out by individual scientist as part of their own work plans (Figure 3-1).

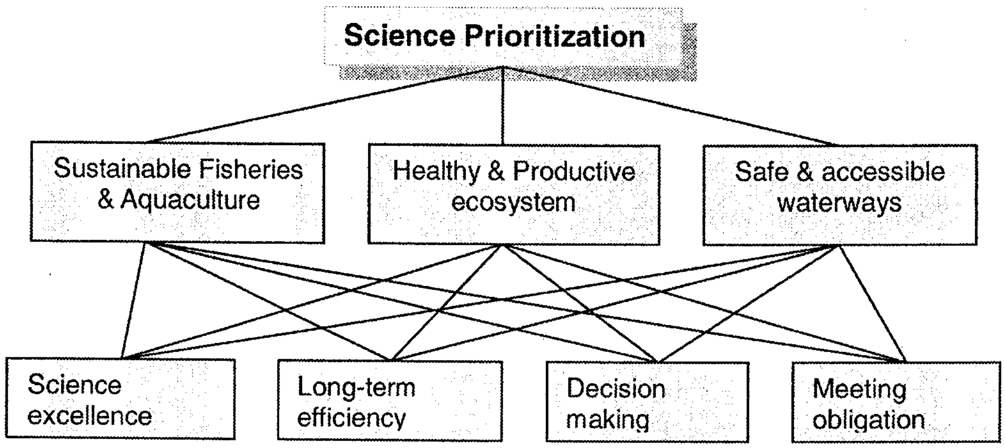
Figure 3-1: The delivery hierarchy of DFO science



The first level of hierarchy, the National level, embodies the current national perspectives and policy thrusts of the department. Figure 3-2 illustrates a risk-based priority setting model built for science prioritization at the National level. In this hierarchical structure, three proposed strategic outcomes of DFO compose the subgoals under the science prioritization, and other DFO-identified "risk areas" are considered as the multiple criteria/attributes in the AHP model (DFO 2003b).

The alternatives for this National level hierarchy can be set differently depending on the problem formulations chosen by decision makers. For example, as for institutional problem formulation, all regions under DFO headquarters should be defined as the alternatives, but as for functional problem formulation, different science service carried out by DFO science should be considered as the alternatives instead. The detailed formulation is explained in the following section.

Figure 3-2: National level hierarchy of DFO Science



The second level hierarchy is built within each individual Region. There are six regional offices under the DFO headquarters in Ottawa according to the Fisheries and Oceans 2003-04 Report on Plans and Priorities (Canada 2003). These include:

1. Newfoundland Region, the Newfoundland and Labrador Region.
2. Maritimes Region, extends from the northern tip of Cape Breton to the New Brunswick-Maine border.
3. Gulf Region, extends from the New Brunswick-Quebec border to the northern tip of Cape Breton with its headquarters in Moncton, New Brunswick.
4. Laurentian Region, falls within the borders of the province of Quebec, and encompasses 6000 kilometers of coastline.
5. Central and Arctic Region, conducts programs to conserve and manage fishery resources for sustainable use in Nunavut and the Northwest Territories.
6. Pacific Region, oversees west coast marine resources and the inland

fisheries of the Yukon Territory.

These Regions all contribute to the national goal. They each also have their own perspective toward regional strategic objectives. The Regional hierarchy should be built based on these objectives and thus characterizes Regions differently.

The third level hierarchy is built for each Division under different Regions. Within each Region, the Divisions deliver and take charge of their own science programs, which are the results of specific science Projects within each Regional Program. Each Division performs toward divisional objectives and contributes to the ultimate goal of DFO science. The Divisional hierarchy represents these objectives individually and appropriately. For example, the hierarchies for Divisions under Maritimes Region including Marine Fish, Diadromous Fish, Invertebrates Fisheries, Marine Environmental Science, Oceans Science, Aquaculture, and Canadian Hydrographic Service represent their own objectives that contribute to the strategic target of Maritimes Region.

Finally the program level hierarchy can be constructed under the programs in Divisions. Different science program have their specific goals and work to deliver the ultimate goal of DFO. For this level's hierarchy, the science project carried out by individual scientist constitute the final level alternative.

As mentioned before, the goal of the science prioritization exercise is to develop a procedure to assign weight to the science programs at each level of the hierarchy, and ultimately enable the resource allocation via these weights down to the level of Projects. There are potentially different objectives hierarchy within each organization level for each Region, and accordingly various programs and projects pursuing these regional objectives. To maintain the hierarchical nature and ensure the consistency and efficiency of the operation of organization, it is necessary to apply a suitable method for integrative science priority setting, where the higher level goals can be met by the lower level components' objectives. All alternatives of each level's hierarchies have minimum budget requirement. The methods used here mimic the actual process of DFO resource allocation, which is not an optimization process but more of a bargaining and negotiation process.

Based on the application of past approaches, the multicriteria weighting method of AHP can be applied to take into account the varied hierarchical expressions of the science objectives.

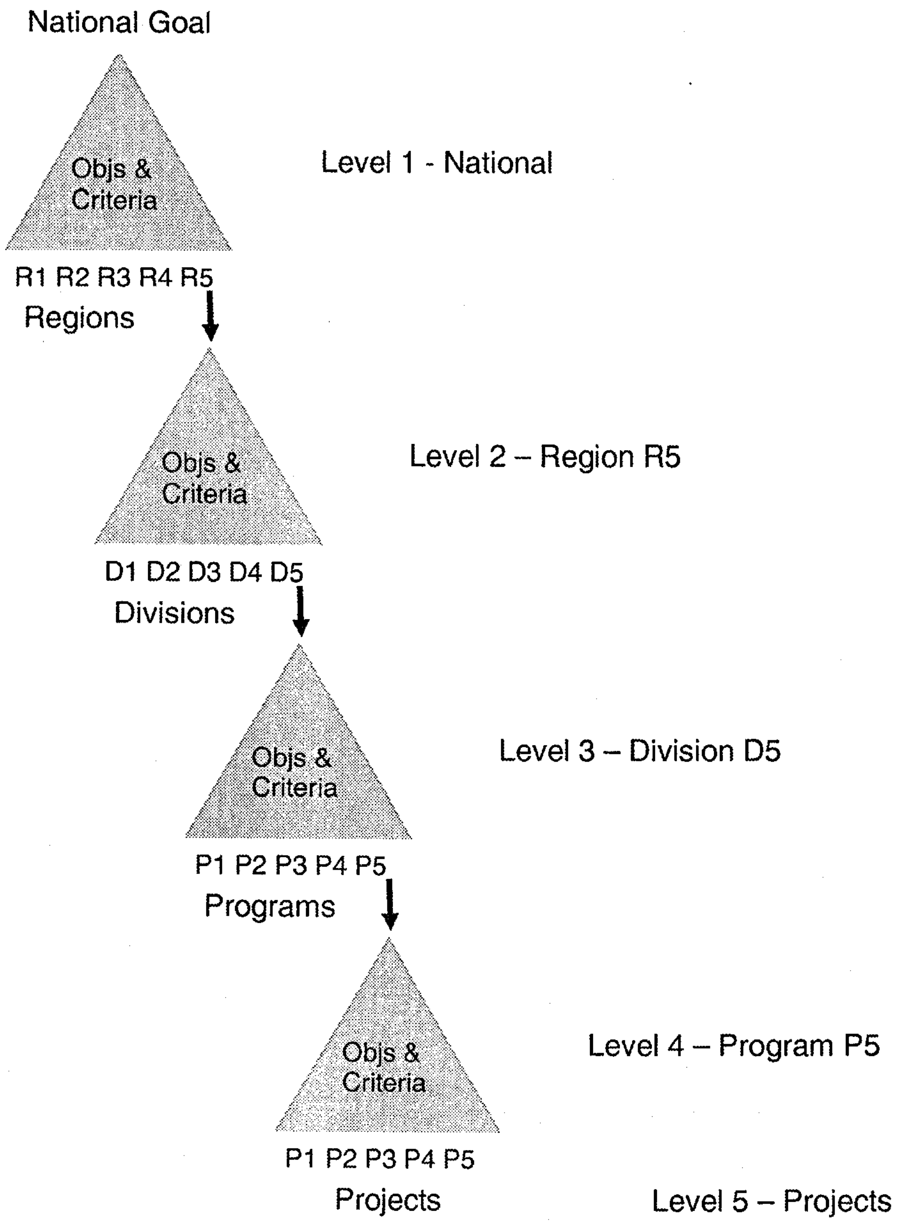
In the following sections, different models depending on the integrity of the science program and the organizational similarities and differences of DFO will be developed to illustrate an approach for helping national priority setting for science programs and activities, and linking science resources to departmental and government priorities. The methods using here will demonstrate that science prioritization decisions and allocations may be based on rigorous structured approach instead of on rounds of negotiation, intuition or anecdotal justification.

3.3 Institutional Problem Formulation

The DFO organizational arrangement model requires prioritization take place at each level of the organization with a filtering down of the information ultimately to the lowest Project level. Figure 3-3 formulates DFO's science priority problem for a specific region R5 applying the principles of allocation from the top down according to the institutional organization. It displays the linked hierarchies associated with the Institutional View problems (Lane 2003).

At each level of the problem (National, Regional, Divisional, and Program), a multicriteria problem is defined that requires a weighted ranking of the allocation to be made at each level. For example, suppose that the total 2003-2004 DFO Science Budget is established as $250 million. This first problem is to allocate these monies to each of the Regions. This is carried out by constructing a Goal, Objectives (or subgoals), and Objectives Criteria in a hierarchy. At the first level, the Regions are designated as the alternative recipients. A structured pairwise comparison is then carried out on: (i) the Objectives, (ii) the Criteria, and (iii) performance or capability of each Region against the Objectives and Criteria combinations. The end result is a weighted ranking of the Regions that evaluates the capability of each Region to contribute to the overall National Goal. A Region evaluated as 0.50 would be allocated 50% of the $250 million allocation or $125 million to be reallocated within the Region.

Figure 3-3: Institutional problem formulation for Region R5 in DFO Science



The second level problem is carried out within each Region. Given a budget (from above), each Region may independently evaluate and allocate its budget within its own Divisions. Arguably, this Regional autonomy should proceed with a direct commitment that the National objectives and criteria are to be attained. However, this can be achieved through the second level Regional Goal, Regional Objectives (or subgoals), and Regional Objectives Criteria in a hierarchy are compatible with (or even identical to) the National problem hierarchy. The result of pairwise comparison would provide weighted rankings to the Divisions and resulting allocations.

The third level problem is a within Division problem then directs allocation from this stage down to Programs. As for the Regional-Divisional problem, the allocation problem of Divisional resources to Programs can be viewed as an independent problem with its own specific goals, objectives, and criteria. The linkage back to National goals could be made more or less explicit. Weighting results direct resources and budget at the Division level down to the Division Programs.

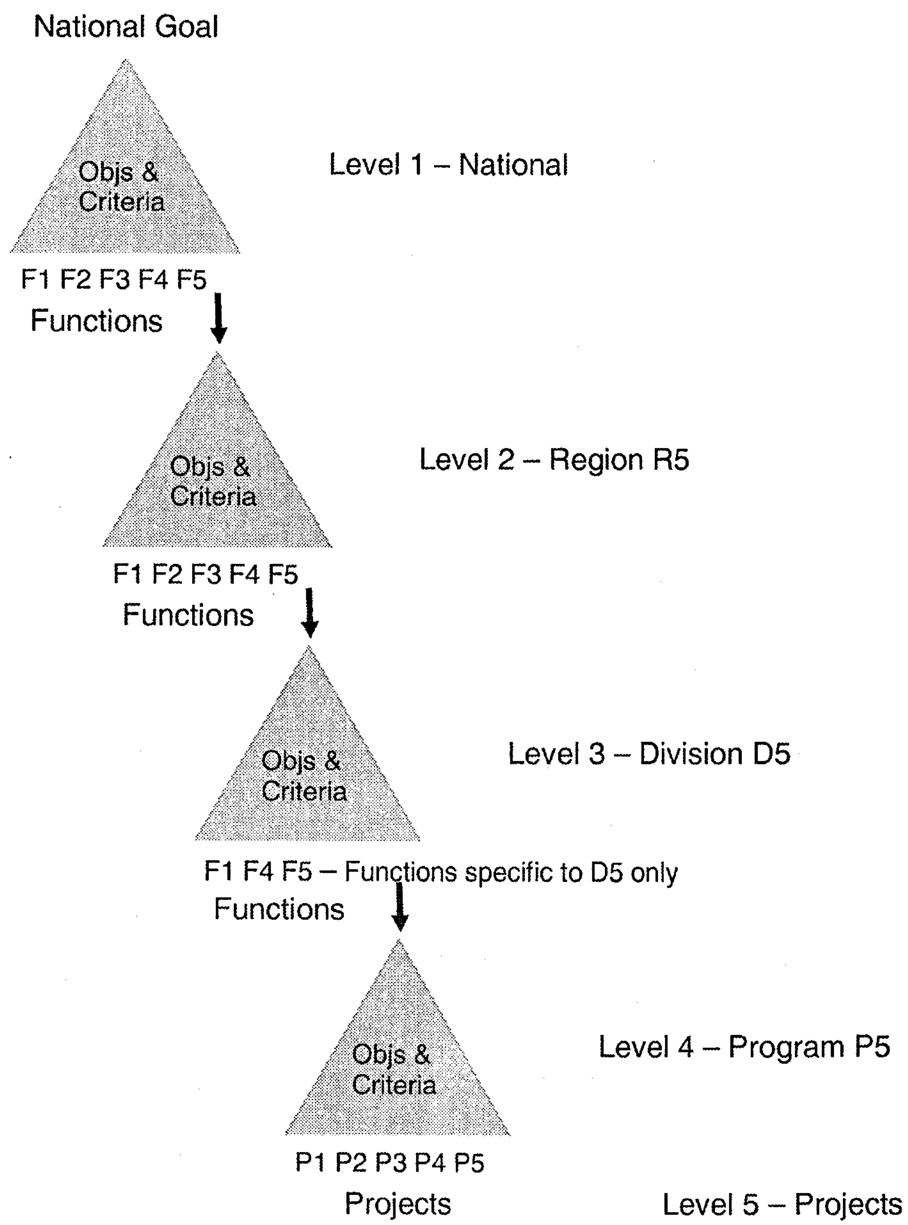
The fourth and final level problem allocates resources of each of the Divisional Programs (allocated resources and budget from the stage above) to the specific Projects within each Program. At this level, it may be required to adjust and rescale Projects in order to compensate for the resources forthcoming from the weighting allocations and for the demands imposed by the proposals of individual scientists work plans (Lane 2003).

3.4 Functional/"Business Line" Problem Formulation

This model requires prioritization take place at each level of the four levels (as presented above), as before, however, the attempt here is to take advantage of a homogeneous application of science services that may be assumed to applied to the organizational units. These homogeneous science services may be described in terms of the DFO Business Line science functions, e.g., Fisheries and Oceans Science, Hydrography, Habitat Management and Environmental Science. In this model it is generally assumed that resources and budgets are more or less fixed for each level of the problem (e.g., the budget for each Region, each Division, and each Program are determined based on some other criteria such as existing size, manpower, etc.). The problem remains as to how to allocate these budgets within each level hierarchy (Lane 2003).

Figure 3-4 formulates DFO's science priority problem for a specific region applying the principles, goals and attributes to all levels of the problem from the top down. It displays the linked hierarchies associated with the Functional View problems. As before, at each level of the problem (National, Regional, Divisional, and Program), a multicriteria problem is defined that requires a weighted ranking of the allocation to be made at each level.

Figure 3-4: Functional /"Business Line" sample problem formulation of DFO Science



Suppose again that the total 2003-2004 DFO Science Budget (for science services) is established as $250 million. This first problem is to allocate these monies to each of the services on the National level. This is carried out by constructing a Goal, Objectives (or subgoals), and Objectives Criteria in a hierarchy. At the first level, the science Business Line functions are designated as the alternative recipients. A structured pairwise comparison is then carried out on: (i) the Objectives, (ii) the Criteria, and (iii) performance or capability of each service against the Objectives and Criteria combinations. The end result is a weighted ranking of the services that evaluates the importance of each service to contribute to the overall National Goal. A service evaluated as 0.50 would receive 50% of the $250 million allocation or $125 million to be reallocated nationally, i.e., distributed to all Regions.

The second level problem is carried out within each Region. In this model each Region is tasked to evaluate its capability relative to the same business line functions as those evaluated nationally. This problem is defined through the Regional Goal, Regional Objectives (or subgoals), and Regional Objectives Criteria in a hierarchy that is compatible with (or even identical to) the National problem hierarchy but viewed from the Regional's perspective and capabilities. The result of pairwise comparisons would provide weighted rankings to how the Region allocates its functions along business lines.

The third level problem is a within Division problem then directs allocations down to a subset of the National business lines Programs. In other words, Regional Divisions are assumed to be already defined along the functional business lines. Thus, the National business lines Programs are separated into the appropriate Divisions and are thereby evaluated with ranking weights via pairwise comparisons.

The fourth and final level problem allocates resources of each of the Divisional Programs (allocated resources and budget from the stage above) to the specific Projects within each business line Program. At this level, it may be required to adjust and rescale Projects in order to compensate for the resources forthcoming from the weighting allocations.

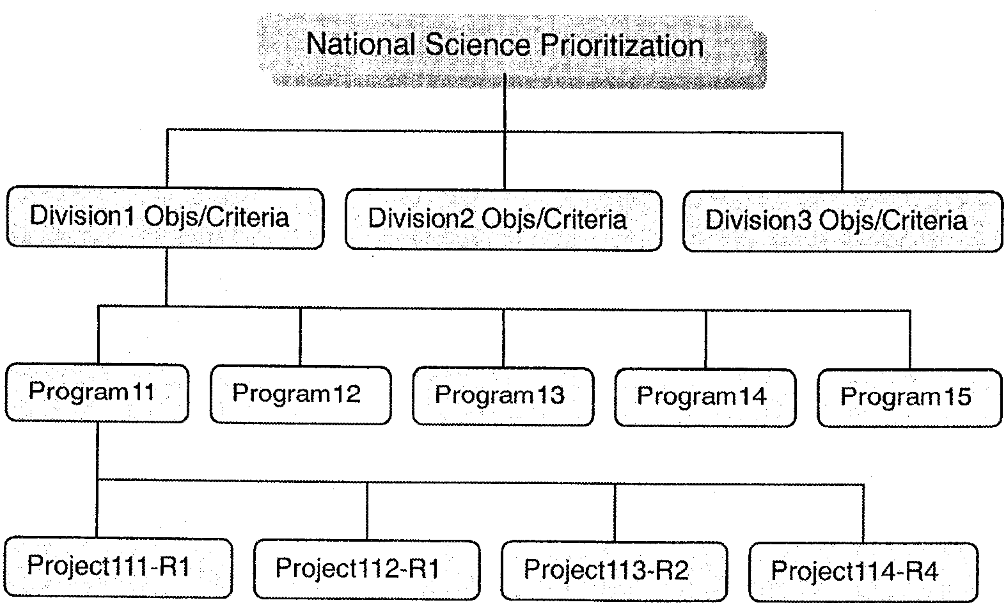
In these two methodologies discussed above, we consider DFO science prioritization as a level model. From national level to the project level, it is assumed that all programs and projects are defined and they are generally assumed to be mutually exclusive and exhaustive. Projects that cut across programs or programs that cut across divisions are considered independently within the analysis of each level (Lane 2003).

3.5 Multidisciplinary Problem Formulation

From the multidisciplinary view, the DFO science prioritization problem is treated as a "one big hierarchy" structure. When building the AHP model, under the goal of this hierarchy of science prioritization, we construct elements within the lower level into a comprehensive objective hierarchy where the objectives of the lower level regional institution form the sub-objectives in this objective hierarchy, and all the sub-objectives make their own contribution toward the ultimate goal of the system, instead of treating lower level components as autonomous sub-models. In the bottom of the hierarchy, the different projects are presented as alternatives working and contributing under this "one big hierarchy" toward the science prioritization objective.

For the DFO sample discussed in section 3.4, instead of building components into separate level models according to their position, this method constructs all elements under each region as one hierarchical model shown in Figure 3-5.

Figure 3-5: Multidisciplinary problem formulations sample for DFO Science



As before, first at National level of the problem, a multicriteria problem is defined that requires a weighted ranking of the allocation to be made at. Suppose again that the total 2003-2004 DFO Science Budget (for science services) is established as $250 million. This problem is to allocate these monies from the National level down to each project. This is first carried out by constructing a Goal, Objectives (or subgoals), Objectives Criteria and alternative Programs and Projects in one comprehensive hierarchy. The highest level of this hierarchy is the same Goal as before, science prioritization. The following levels are the objectives (or subgoals) of each Division. Under every Division, there are various Programs working toward the Objectives setting for the Division, and within each Program, there are different Projects which will be carried out by scientists in different Regions. In this model, Regions are designated as part of the properties of each project in the lowest level instead of being actually a resource allocation level. For example, under Division1, there are many Projects proposed to achieve the objective of Program11. Project112-R1 refers to the 2nd project, which is proposed by Region1, under Division1 and Program11.

A structured pairwise comparison is then accomplished on: (i) the Objectives, (ii) the Criteria, and (iii) performance or capability of each Project against the Objectives and Criteria combinations. The end result is a weighted ranking of the Projects that evaluates the importance of each Project's contribution to the overall National Goal. Finally, the configurations of the allocation to the Region are constructed from their participation in the mix of regional projects accepted form allocation.

3.6 Risk Analysis for DFO Science Prioritization

The aggregation of risk in the problem formulation of decision making process entails a commitment to a more structured, transparent and consistent approach to set priorities based on the significance and likelihood of uncontrollable outcomes. The contents of this section embrace the risk analysis approach discuss in general term in section 2.3 above into the DFO risk-based science prioritization model and practice.

3.6.1 Application of Risk Analysis and Risk Assessment

As for the situation in DFO science prioritization, risk analysis can be interpreted in terms of two components: risk assessment and risk management (Lane and Stephenson 1998).

- Risk assessment - the process that evaluates possible outcomes or consequences and estimates their likelihood of occurrence as a function of the uncontrollable state dynamics of the system.
- Risk management - the process whereby decision makers use information from risk assessment to evaluate and compare decision alternatives.

For many research scientists, "risk analysis" constitutes the exercise of determining the probabilities of ecological output measures that indicate failure only. In this research, this exercise is more appropriately identified as one part of the "risk assessment" process. In the context of past management decision making problem in DFO, the risk assessment process has been limited to estimating the probability that the outcome of a given management decision will exceed some predefined management thresholds. However, in the context of expected utility analysis introduced here, the ranking of the decision alternatives is based on the expectation over the entire probability distribution of outcomes. To limit the estimation of probability to purely undesirable outcomes is not complete in terms of risk analysis because risk associated with the prioritization problem must be reflected appropriately over the whole output range.

For example, consider the DFO science prioritization problem which needs to allocate a certain budget between two science programs A and B. Expected contribution is used as an indicator of a satisfactory level of individual program outcome with respect to the ultimate goal of DFO science. Suppose the target contribution for each program is set to 100%, and the costs to implement these programs are similar. Program A achieves 40% contribution with probability 0.7 and the probability is 0.3 that the undesirable outcome appears with no contribution at all. Program B achieves 80% contribution with probability 0.6 and the probability is 0.4 that the undesirable outcome occurs during the implementation process. The probability distributions for the two programs are shown in Table 3-1 and Figure 3-6.

Table 3-1: Probability distribution of the outcome of utility analysis sample

Programs \ Outcomes	Success		Fail	
	Probability	Contribution	Probability	Contribution
A	0.7	40%	0.3	0
B	0.6	80%	0.4	0

The problem is to make a decision on program selection. If we simply make the decision based on the probability of undesirable outcome ("A_f" and "B_f" in Figure 3-6), the selected program would be program A since the probability that A fails is less than that of B. However, this decision ignores the information about the contributions if the programs would be successful. As shown in the Figure 3-6 ("A_s" and "B_s"), the contribution from program B is twice as large as program A when the programs are successful. Therefore the expected value of B is calculated as:

$$E_B = 0.4 \times 0 + 0.6 \times 0.8 = 0.48 \tag{22}$$

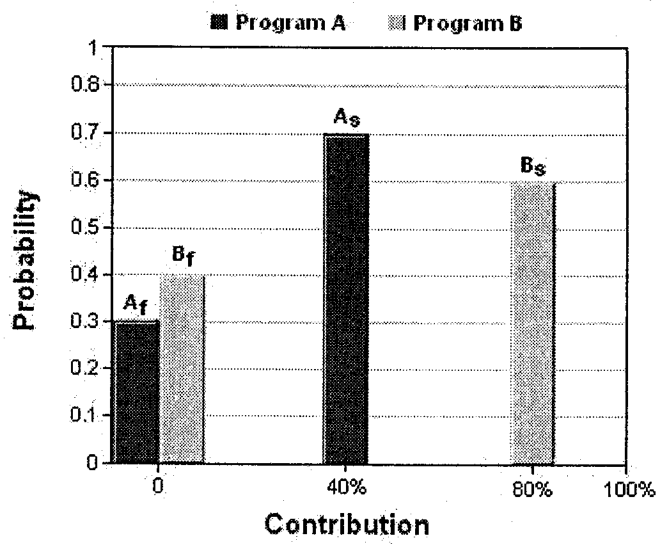
The expected value of A is calculated as:

$$E_A = 0.3 \times 0 + 0.7 \times 0.4 = 0.28 \tag{23}$$

Accordingly, taking into account all program outcomes, the budget should be allocated to selected program B based on the expected value of the individual programs.

Therefore the decision based on the probability of the unwanted failure event happen is only one of the performance measurements of program, and is inappropriate to be used for risk analysis in prioritization problem. The risk analysis process for decision making under uncertainty should consider all possible outcomes with uncertainties so that the full evaluation is properly taken into account. In this thesis research, proceeding under a unified and consistent approach integrating the concepts of risk, uncertainty and utility discussed in previous chapter, risk assessment is only the first part of the risk analysis in the DFO risk-based science priorities setting process. The resulting priorities ranking of these model simulations is then used to provide information to a further stage of the process, namely risk management, to evaluate and compare different alternatives toward final ranking and decision support.

Figure 3-6: Probability distribution of the outcome of utility analysis sample



3.6.2 Risk Management

For risk management, also referred to in formal decision analysis as "utility analysis", clearly defining "objective performance criteria" as a means of measuring the success and failure of science programs and providing accountability within the prioritization process is crucial (Bertsekas 1976). The risk management of the DFO risk-based science prioritization exercise therefore embodies the decision making criteria in a utility function that measures the expected performance of each alternative in terms of the multiple criteria and their trade-offs, and consequently evaluates and ranks the alternative for decision makers.

According to the approach to risk analysis described above in Chapter 2, before the problem can be solved numerically, the form of the utility function must be specified. The utility function used in this thesis is defined experimentally using the S-shaped Utility Curve shown in Chapter 2. To further simplify the simulation process, piecewise linear estimation is used to actually calculate the utility with respect to the budget allocated to individual alternatives.

To illustrate the formulation of program utility function, consider the utility curve determined by the critical points

$$(x_i, u_i), \quad i = 1,2,3,4,5 \tag{24}$$

where x_i is the dollar value indicating the budget allocated to program

u_i is the utility corresponding to budget allocation x_i.

Setting the scales u_i experimentally as $u_1 = 100\%$, $u_2 = 80\%$, $u_4 = 50\%$, $u_3 = 20\%$, $u_5 = 0\%$, defines the critical points as follows:

1. (x_1, u_1): Maximum Anchor, the performance of program achieves 100% satisfaction with the budget allocation x_1.
2. (x_2, u_2): Target Threshold, the performance of program reaches 80% satisfaction with budget amount x_2
3. (x_3, u_3): Mid-range Threshold, the performance of program achieves 50% satisfaction with the budget allocation x_3.
4. (x_4, u_4): Limit Threshold, the performance of program achieves 20% satisfaction with the budget allocation x_4.
5. (x_5, u_5): Minimum Anchor, the performance of program achieves 0% satisfaction with the budget allocation x_5.

Assume that the resulting budget allocation to the program from AHP prioritization process is the threshold of the program, which means that the utility of the program under the current budget is 50%, and then the corresponding x_i values are set to be calculated initially by the function

$$x_i = \frac{(B \times P) \times \alpha_i}{\alpha_3} \qquad i = 1,2,3,4,5 \tag{25}$$

where $B =$ Total budget

$P =$ Program proportion of budget

$\alpha_i =$ Set of scalars, where $\alpha_1 = 1, \alpha_2 = 0.8, \alpha_3 = 0.5, \alpha_4 = 0.2, \alpha_5 = 0$

Now suppose the total budget is $B = \$350{,}000$, and the resulting budget proportion of a specific program is 20%, then the values and definitions of critical points for the s-shaped utility curve of this DFO science program are listed in Table 3-2. Accordingly, the piecewise linear estimation function shown in Table 3-3 is used to calculate the utility with respect to a specific budget allocation, where x denotes the budget allocation, and $U(x)$ is the utility value corresponding to x.

Table 3-2: Value definition of sample utility curve

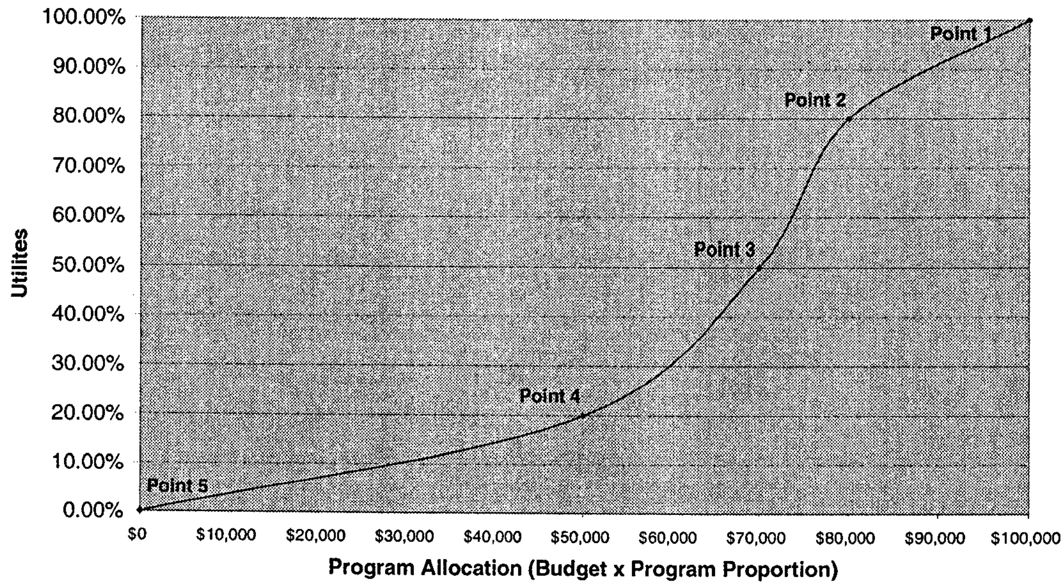
Point	Point No.	Measure	Utility	Risk Description at Point
Maximum Target Anchor	1	$100,000	100.00%	Risk Averse
Target Threshold	2	$80,000	80.00%	Risk Averse
Mid-range Threshold	3	$70,000	50.00%	Risk Neutral
Limit Threshold	4	$50,000	20.00%	Risk Seeking
Minimum Anchor	5	$0	0.00%	Risk Seeking

Table 3-3: Piecewise linear estimation of sample utility curve

Points	Budget (x)	Intercept	Slope	Linear Function
1-2	$\$100{,}000 \le x \le \$80{,}000$	0.000000	0.000010	$U(x) = 0.00001x$
2-3	$\$80{,}000 \le x \le \$70{,}000$	-1.600000	0.000030	$U(x) = 0.00003x - 1.6$
3-4	$\$70{,}000 \le x \le \$50{,}000$	-0.550000	0.000015	$U(x) = 0.000015x - 0.55$
4-5	$\$50{,}000 \le x \le \$0$	0.000000	0.000004	$U(x) = 0.00004x$

Finally the resulting s-shaped utility curve for this specific program is shown in Figure 3-7. It is noted that different participants faced with the same prioritization problem may have different utility functions, so that the critical points which decide the shape of utility function should be decided by participants during the risk analysis process. The method for calculating the value of critical points for the s-shaped utility curve discussed above is only a set of experimental functions to initialize the utility function. This initially calculated utility function for individual alternatives is not fixed and considered to be the input for risk analysis in the risk-based science priorities setting procedure.

Figure 3-7: S-shaped curve for sample of utility analysis



3.7 Procedure of DFO Science Prioritization

Based on the literature review of methods for science prioritization and the situation analysis of DFO above, a proposed procedure designed specifically for DFO case is described in this section. By characterizing and comparing the current process of DFO science resource allocation, this thesis research proposes a rigorous approach to promote a scientific decision making process, which is more transparent, participatory, and informed through sound science.

3.7.1 Current Process

Each year, the government prepares Estimates in support of its request to Parliament for authority to spend public monies. The Estimates consist of three parts including: (1) The Government Expenditure Plan providing an overview of federal spending; (2) Main Estimates providing a detailed listing of the resources required by individual departments and agencies for the upcoming fiscal year; (3) Departmental Expenditure Plans, which is divided into two components, Reports on Plans and Priorities (RPP) providing individual expenditure plans for each department and agency and Departmental Performance Reports (DPR), individual department and agency accounts of accomplishments achieved against planned performance expectations as set out in respective RPP (Canada 2002a).

The RPP at Fisheries and Oceans provides increased levels of detail on a business line basis and contains information on objectives, initiatives and planned results, including links to related resource requirements over a three-year period. The RPP accurately portrays the organization's plans and priorities based on sound underlying departmental information and management systems. It covers the planned spending information which is consistent with the directions provided in the Minister of Finance's budget and by Treasury Board Secretariat. The process of preparing this document can be simplified into three stages (Canada 2003).

The first stage, corresponding to the long term strategic targets of DFO, defines the objectives and multiple criteria of DFO science prioritization into a hierarchical structure from department top-down to projects on a business line basis. The Department has 11 business lines, with seven Assistant Deputy Ministers (ADMs) responsible for the 11 business lines. The ADMs are responsible for establishing national objectives, policies and procedures, and standards for their respective business lines. This is done in the cooperation with the regional participations. Within this structure building stage, programs along the business line are compiled using the Department's accountability structure as approved by Parliament. Planned spending and expected contribution of programs and projects are listed along with their descriptions (Canada 2003).

The second stage, determine the importance of each project through negotiation with management officers. This negotiation is an indispensable process for the decision making. The Department's client base has also been evolving, and increased public and stakeholder expectations are placing pressure on its ability to meet its regulatory responsibilities effectively. In the face of more transparent communication, consultation has become critical in addressing the concerns of diverse stakeholders. Also DFO must ensure that it is in compliance with international agreements and consider the impact of its activities on the international community (Canada 2003).

The third stage, according to the results of the discussion in the second stage negotiations, implements resource allocation decisions for DFO science operations from the department top-down to each region, division, program and project. Fisheries and Oceans Canada operates across Canada from six regional offices, as well as national headquarters in Ottawa. Each of the six regions is responsible for organizing and managing the delivery of programs and activities in their regions in accordance with national and regional priorities and with national performance parameters set for each program and activity. In short, their role is to mobilize the process and translate the strategic direction into actions at the field level (Canada 2003).

3.7.2 Proposed Process

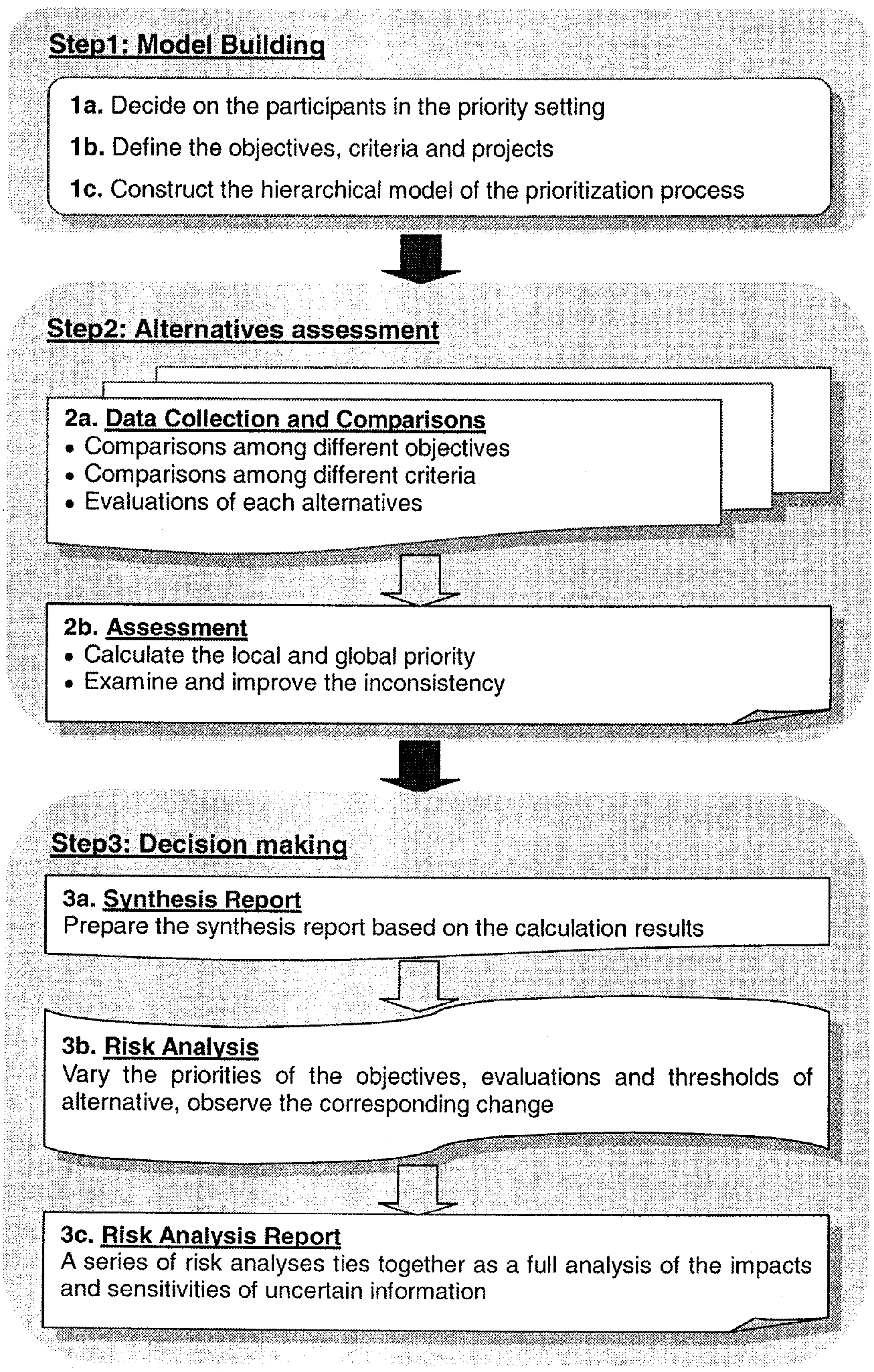
The process proposed for allocating science budgets in this thesis research is designed based on the multicriteria framework of the AHP model construction procedure for science prioritization. The specific steps for the DFO case are introduced as follow and also shown in Figure 3-8 (ISNAR 2001):

Step 1. Model building including three sub-steps described below:

1a. Building the right team is the first requirement for effective priority setting. Priority setting builds on the knowledge, opinions and judgments of many having different expertise. Within the governance requirements of the DFO science operation, the department needs to assemble a group of people that are able to represent the most important interests and allow effective interaction and communication.

1b. Defining objectives, criteria and alternatives. A system analysis is an essential prerequisite for a solid priority setting exercise. It reviews the main policies and reforms that influence the performance and constraints of the system and its components. For DFO case, different structure bases are considered for institution view, function view and multidisciplinary view problem formulation as discussed in sections 3.3 to 3.5.

Figure 3-8: Flow chart of AHP for DFO Science prioritization



The science research program will contribute to the overall development process of the country or to the final mission of the organization. Before any further analysis can be undertaken, the research objectives need to be clarified. The projects for which priorities are being defined may contribute to all of these objectives, and ultimately to the goal, science prioritization.

Priority setting is about making choices, but before that, the alternatives among which these choices should be defined. Research alternatives can be identified by analyzing constraints and opportunities, which lead to research project identification, a first characterization of the science prioritizing options.

To assess research alternatives, evaluation criteria need to be formulated. These criteria normally correspond with measurable indicators for the research objectives that were selected earlier. The final selection of evaluation criteria is best made after the research alternatives have been defined. When there are several criteria, criteria weighting may be required.

1c. Building the hierarchical model. The goal, objectives, criteria and the alternatives defined in step 1b are the component for the hierarchical model of science prioritization. Before building the model into application system, each component needed to be positioned into the hierarchy appropriately based on their definition and the interrelation.

Step 2: Alternatives assessment including two sub-steps described below:

With limited resources, research should clearly be undertaken only on those alternatives that contribute most to the research objectives. To choose critically, we will estimate how the research alternatives perform on the measurable criteria defined above by applying the multiple criteria measurement method.

2a. Data collection. The pairwise comparisons data among different objectives and criteria are collected from each participant in the decision making team, where the comparison for criteria as opposed to objectives is made based on each participant's assessment of the relative importance of each criterion. As for alternatives, same procedure can be used to get the pairwise comparison data.

However, it's often the case that the number of alternatives is large enough to make the corresponding matrix calculation become very complicated. Therefore, a datagrid input method can be chosen for evaluating the contribution of each alternative toward every criterion. Instead of using pairwise comparison to evaluate the relative contribution, the datagrid is used to collect the absolute value of the contributions decided by participants based on a set of weight scale.

2b. Assessment. The data collected in 2a is the input of AHP algorithm. As we discussed in the earlier section, through the calculation of eigenvector and eigenvalue, which represent the weight vector of alternatives and the consistency of participant's opinions, the results on each option will be weighted in a rank order from highest to lowest priority.

Step 3: Final decision making which include three sub-steps:

3a. Synthesis report. After all the calculations have been finished, priorities are set in order to be implemented. However, it does not lead automatically to an implementation scheme. Thus the priorities need to be transformed into a resource allocation plan. The resource allocation plan considers what the priorities should imply for the funding of each research alternative, and how the current resource allocation can be modified.

Finally we can arrive at a set of priorities and an allocation of resources that is based on the combined experience of the participants in the exercise, the available knowledge, and a carefully chosen methodology. While all participants may agree with the results, it is still important to obtain further feedback. The process followed and the results obtained up to here should be submitted to wider stakeholder validation. Stakeholder validation is a final step towards improving the quality of the results, and facilitates the implementation of th priority research alternatives.

3b. Risk analysis. Often the initial rank order can be submitted to risk analysis after the initial synthesis report is carried out for the priority setting process, e.g., by means of more rigorous mathematical procedures in which the objective weights and alternative's contributions toward each criterion are modified to examine the effect of such changes, corresponding changes in the priority rank order and resource allocation plan. Moreover, based on the utility theory discussed earlier in Chapter 3, by setting different thresholds for each alternative and changing the total budget, corresponding changes of total utility can be used to support the necessary decision changes.

3c. Risk analysis report. A series of risk analyses from step 3b ties together as a full analysis report on the impacts and sensitivities of uncertain information associated in the alternative's implementation. The report includes results of three components of risk analysis: (1) the revised priorities ranking of alternative corresponding to the changes in the weight of top level objectives; (2) a set of new priorities resulted from the changes in the performance evaluation of individual alternative; (3) utility analysis report using the expected utility approach.

4. Modeling and Analysis

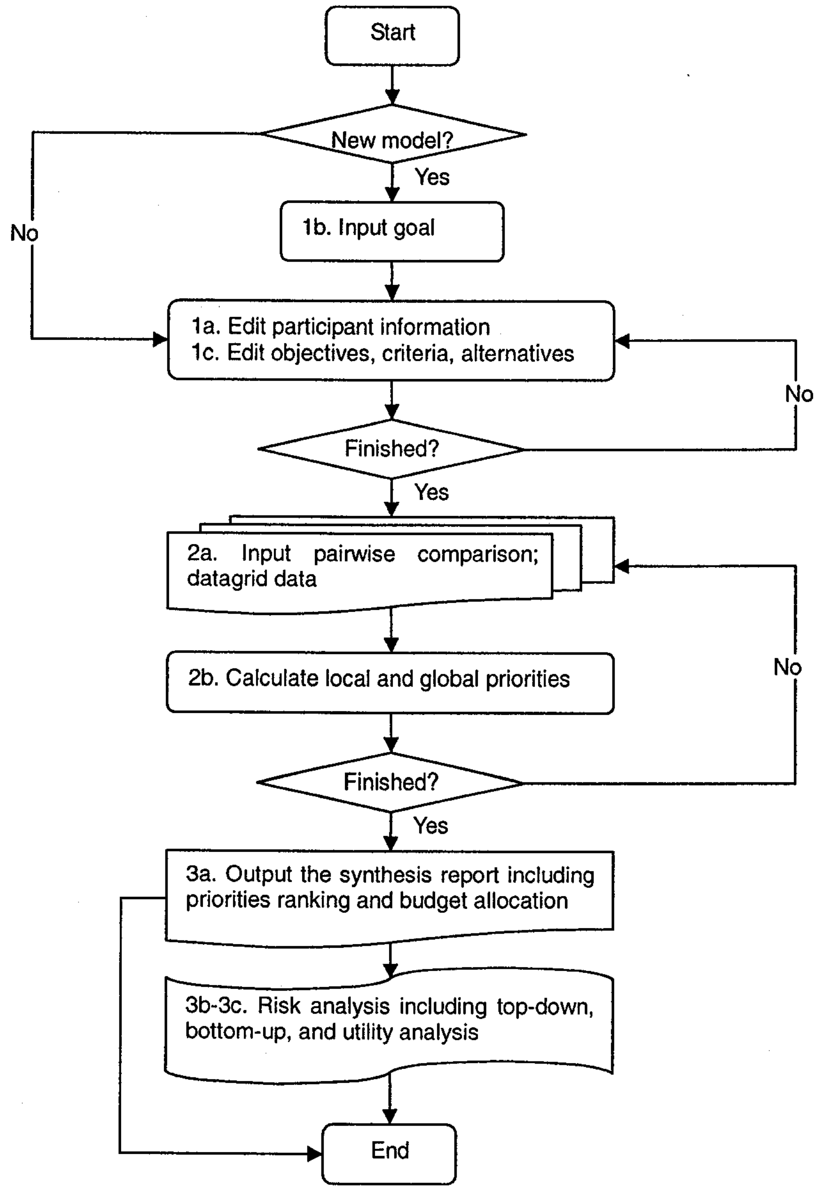
This chapter illustrates a computer-based application system developed to analyze different procedures for the DFO risk-based prioritization problem based on the methodology and approaches discussed above. It introduces modeling strategy responsive to the needs of the DFO science research and decision support activities. Section 4.1 introduces the detail software modeling process of the system along with the technical considerations for model building implementation and illustrative results. Following this introduction, section 4.2 illustrates the model construction specifically for different DFO problem formulations presented in Chapter 3. Finally in section 4.3, the similarities and differences among the results from different problem formulations are discussed.

4.1 Modeling Process

To establish a better decision making procedure, the valid methodology for modeling and simulation of real life system should be chosen reasonably. A successful tool designed for AHP modeling and simulation is Expert Choice software (EC2000), which was first presented in 1983. Based on the Analytic Hierarchy Process (Saaty 1988), EC2000 provides a mathematically rigorous application and proven process for prioritization and decision making.

However, to better fit with the situation in DFO, an application system that can represent DFO risk-based science prioritization specifically is obviously more preferable. The application system in this thesis is developed by Visual Basic.NET (VB.NET) and SQL2000 and based on Windows XP operating system. The system is designed especially for the current research to handle data input, analysis and output reports. Following the proposed process in section 3.7, the system is divided into four relatively independent modules: 1) model construction; 2) assessment; 3) synthesis; 4) risk analysis. The detail process of system will be demonstrated following the system flow chart (Figure 4-1) using a basic AHP model and the detail steps are described in Appendix IV.

Figure 4-1: Flow chart of AHP model simulation

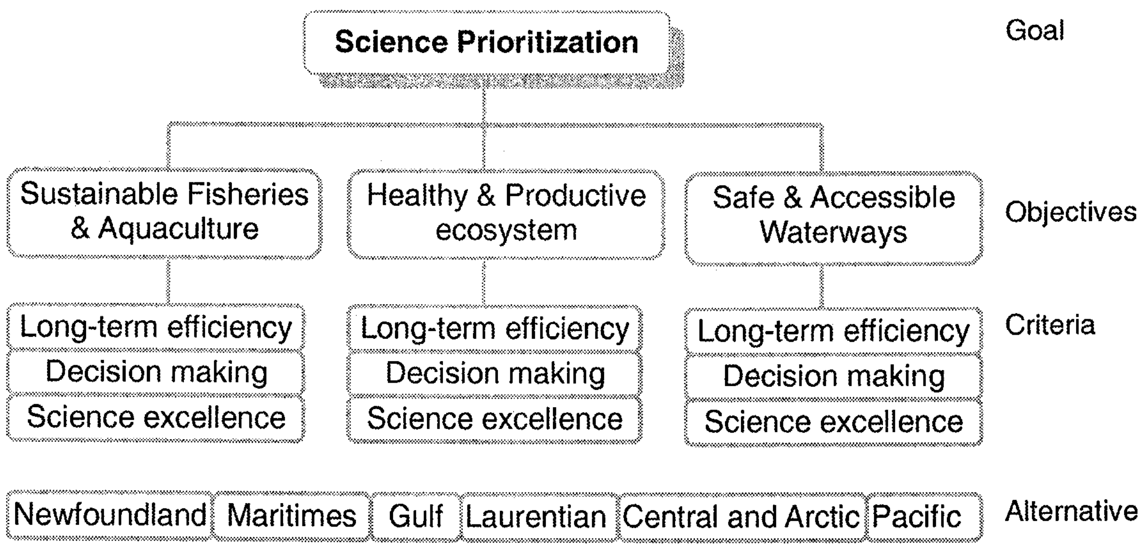


4.1.1 Model Construction

Models are essential tools for synthesizing observations, theory, and experimental results. In DFO case, models can be used in both a retrospective sense, to prove the merit of modeling methodologies by comparing model results with observations of past change, and in a prognostic sense, for calculating the weights of the alternative science programs to projected future strategy. Comprehensive science prioritization models provide the primary quantitative means to integrate scientific understanding of the many components of the DFO science research and, thus, are the principal tools available for making quantitative decisions.

To illustrate the implementation process of AHP model construction especially for DFO case, this section introduces a simplified DFO sample hierarchy shown in Figure 4-2.

Figure 4-2: National AHP model for demonstration of model construction



The purpose of AHP model construction is to decide the priorities for each alternative through rigorous algorithm. The simulation of this AHP sample model follows the steps introduced in section 3.7. This section is corresponding to the step 1 of the proposed process in section 3.7.

1a. Building the right team. This step is about to select the right people for decision making. This group should be large enough to represent the most important concerns and small enough to enable effective interaction and communication. The implementation of this step in the application system is to setup participants for each model during the process of model building. The operation for the participant administration can be done using the function shown in Figure 4-3, and the participant joining the evaluation process of specific model can be setup by a sub-module shown in Figure 4-4.

Figure 4-3: Participant administration of the application system

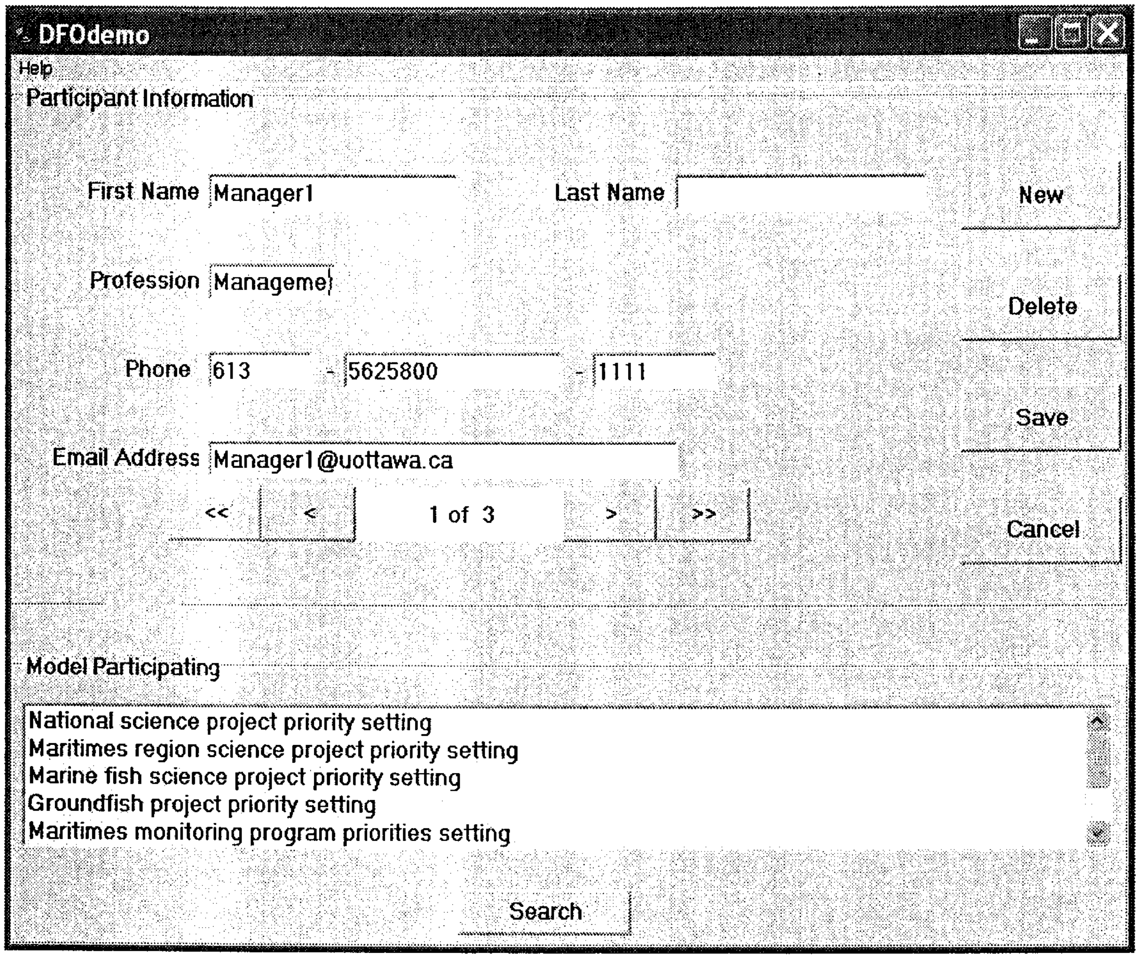
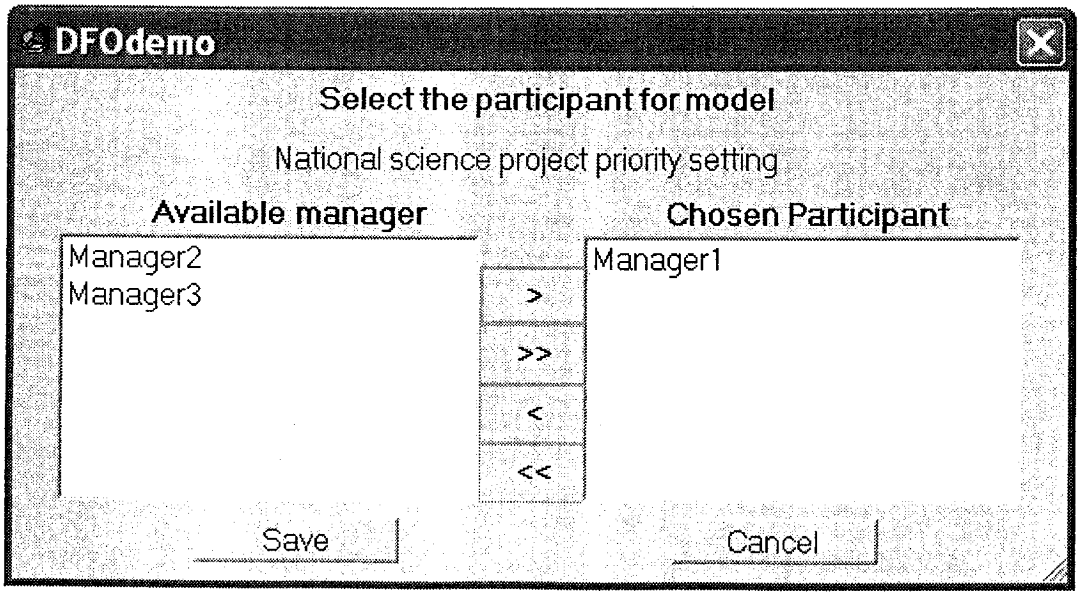
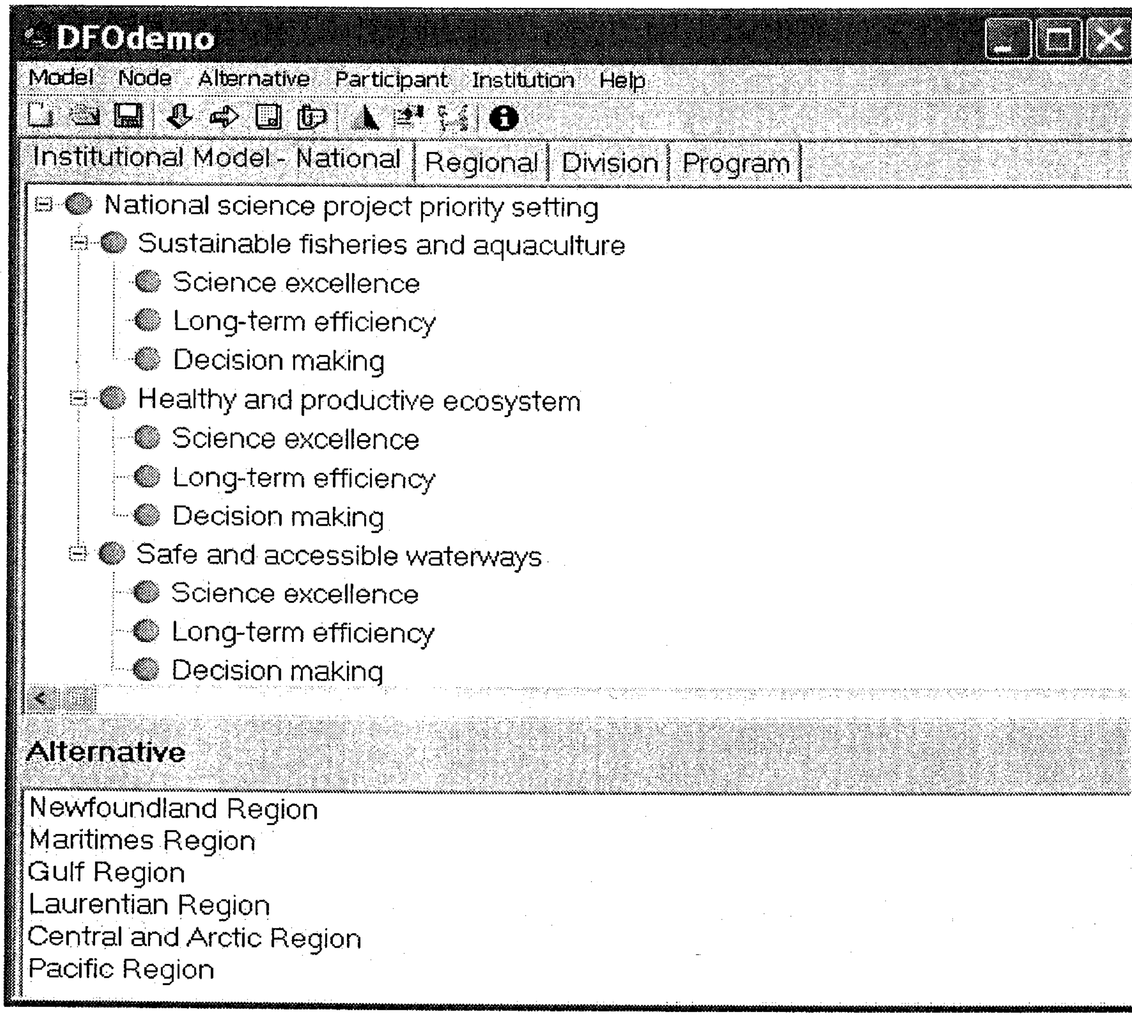


Figure 4-4: Setup participant for specific model in the application system



1b-1c. Model building. The model construction module is the first part of the application system in this thesis. The main function of this module is to actually build the AHP model (Figure 4-2) into the system (Figure 4-5) for the later assess and analysis purpose.

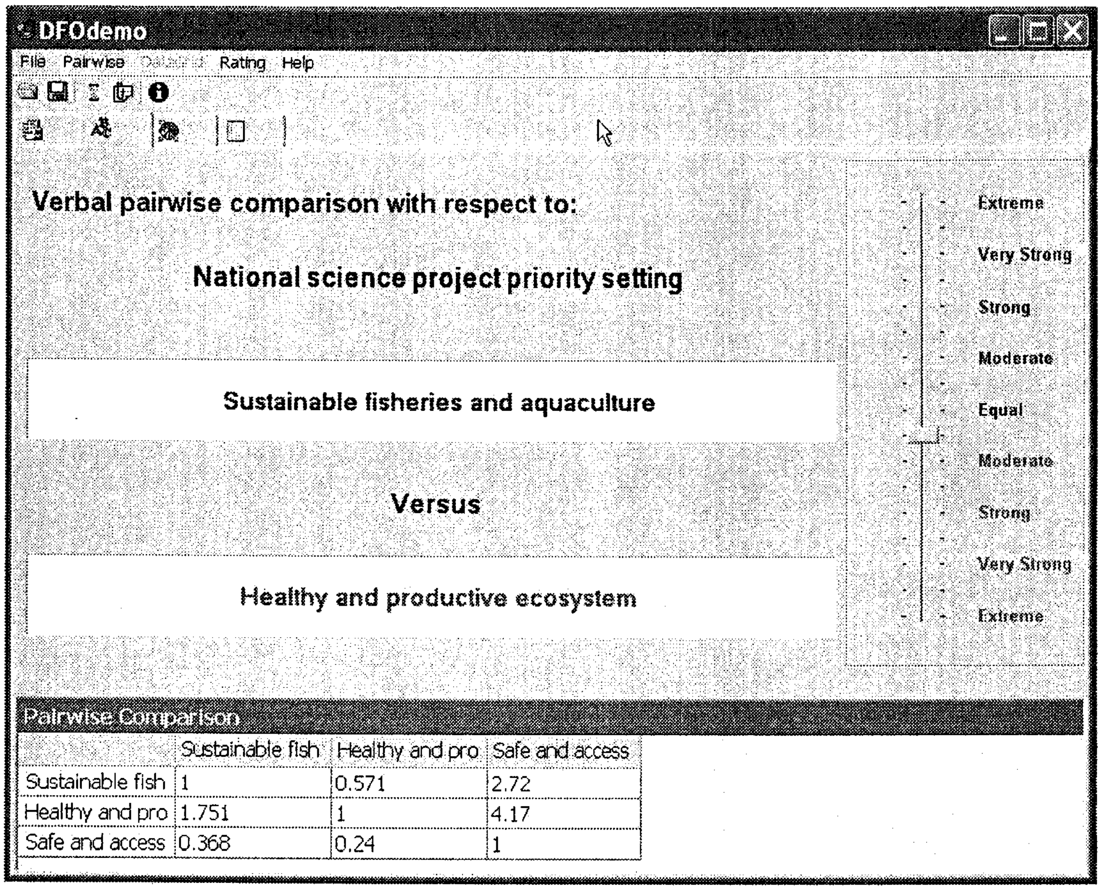
Figure 4-5: Sample model building using application system



4.1.2 Assessment

2a. Data collection. A set of spreadsheet designed based on the model structure can be use to collect the pairwise comparison for objectives and criteria, and the datagrid input for project contributions. A set of sample data collection forms with pairwise comparison data for the sample hierarchy was chosen representing aggregate feedback from a group of six Maritimes Region managers compiled in 2003 following the Oceanstone meeting (Appendix I). The average result of these pairwise comparison results is then input to the system directly or by using the multiple user input function such as the verbal comparison method (shown in Figure 4-6) or graphical comparison method (shown in Figure 4-7).

Figure 4-6: Verbal pairwise comparison of the application system



Since each year DFO resource allocation process need to consider hundreds or even thousands projects proposed by individual scientist, data for the performance of the lowest level alternative toward every objective and criterion will be inputted using datagrid input in the system. A set of rating scale should be set before the participant evaluate the performance of every alternative. A 5-scale rating was setup to suit this purpose by the scale setup sub-module (shown in Figure 4-8). This rating scale system includes the verbal term representing the measurement of project performance, the corresponding scale number as for an easier way to input by participants and the absolute value of the evaluation, which is calculated by scales' pairwise comparison matrix.

Figure 4-7: Graphic pairwise comparison of the application system

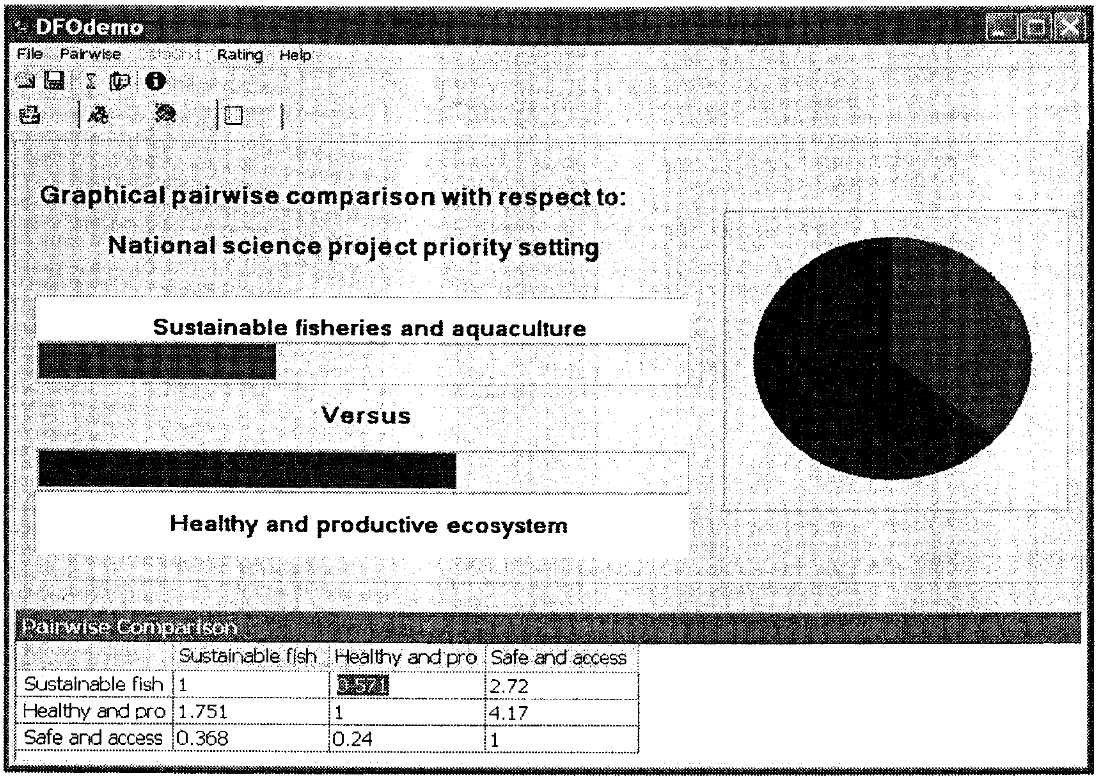
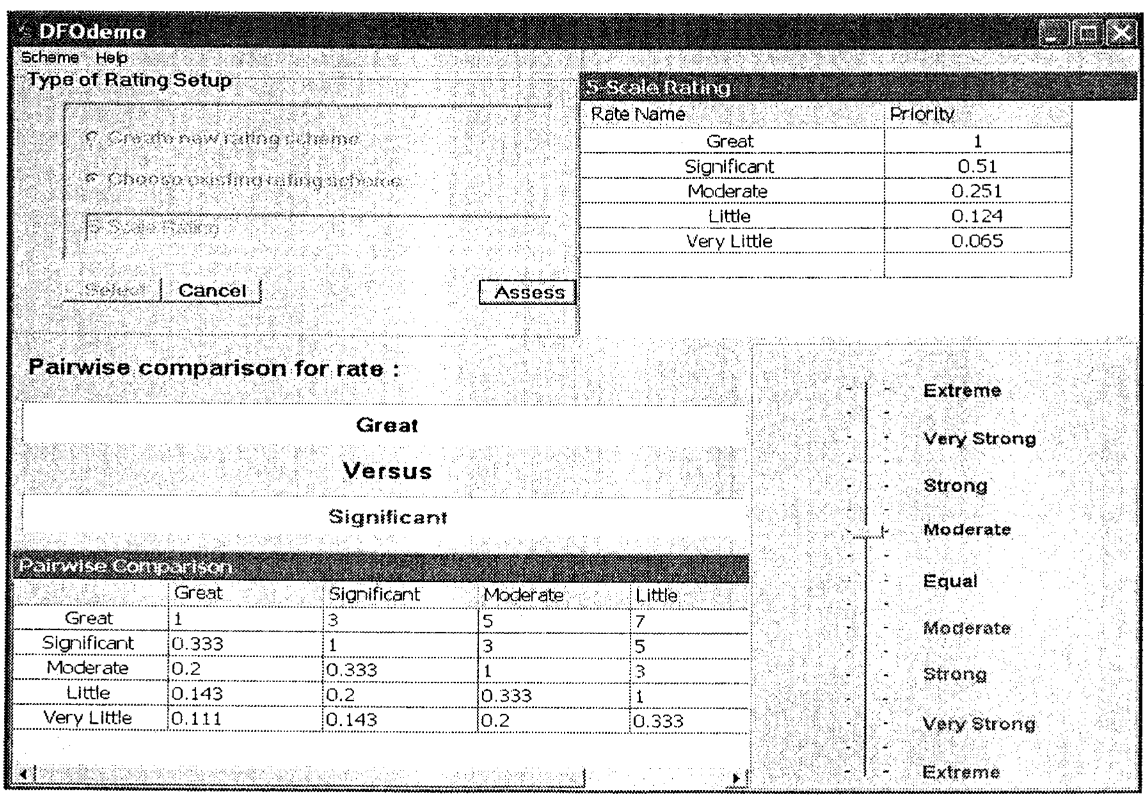
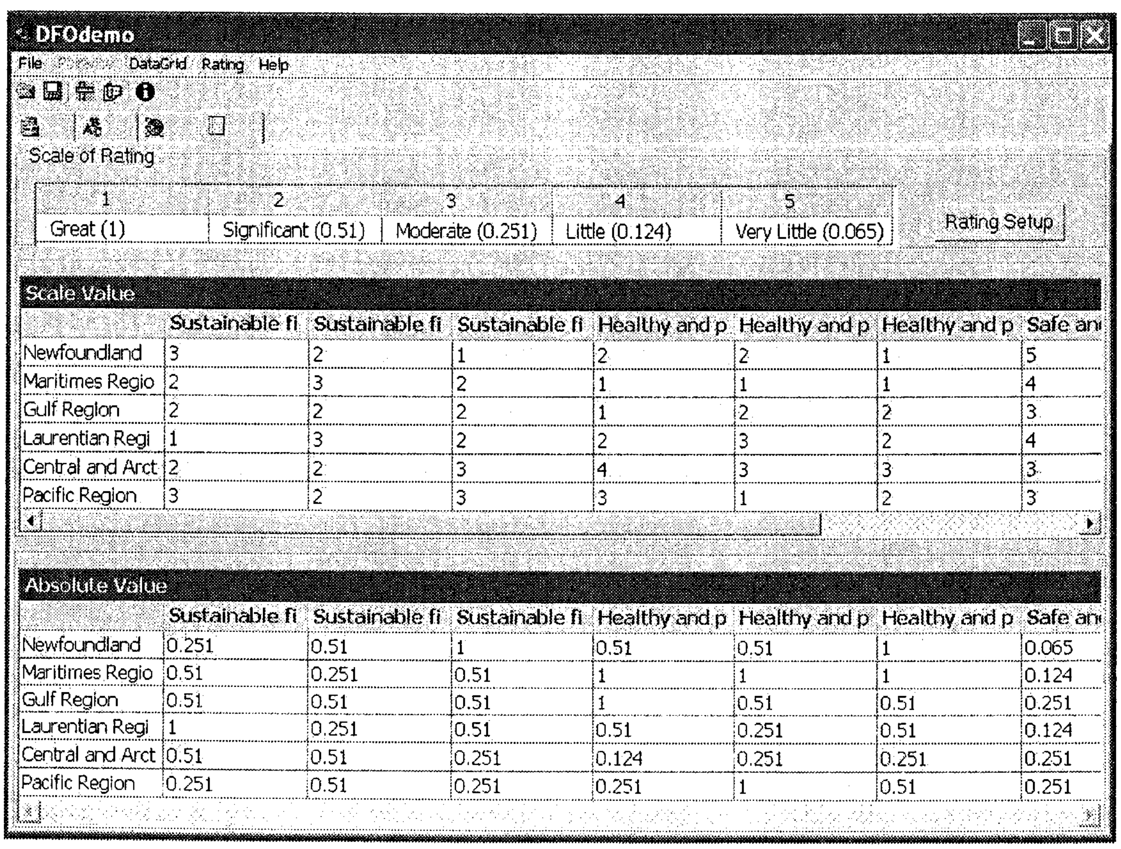


Figure 4-8: Setting rating scale in the application system



2b. Assessment. The outcome of assessment includes the local priorities for objectives and criteria, and global priorities of alternatives. First, the local priorities of objectives and criteria are accomplished by calculating the eigenvector and eigenvalue for the pairwise comparison matrix formulated above. Next, the absolute values of alternatives are changed to values in distributive mode. For example, as for Science Excellence under Sustainable Fisheries and Aquaculture, each alternative has their own absolute value (the corresponding average results from the six managers' pairwise comparison data listed in Appendix I); the distributive mode value of every alternative is then calculated by dividing the individual absolute value by the total of all absolute values (shown in Table 4-1). These distributive values of alternative can be seen as the local priorities with respect to every criterion, and the global priorities of alternatives can be acquired using the algorithm introduced in chapter 2.

Figure 4-9: Datagrid evaluation in in the application system



1	2	3	4	5
Great (1)	Significant (0.51)	Moderate (0.251)	Little (0.124)	Very Little (0.065)

Scale Value

	Sustainable fi	Sustainable fi	Sustainable fi	Healthy and p	Healthy and p	Healthy and p	Safe an
Newfoundland	3	2	1	2	2	1	5
Maritimes Regio	2	3	2	1	1	1	4
Gulf Region	2	2	2	1	2	2	3
Laurentian Regi	1	3	2	2	3	2	4
Central and Arct	2	2	3	4	3	3	3
Pacific Region	3	2	3	3	1	2	3

Absolute Value

	Sustainable fi	Sustainable fi	Sustainable fi	Healthy and p	Healthy and p	Healthy and p	Safe an
Newfoundland	0.251	0.51	1	0.51	0.51	1	0.065
Maritimes Regio	0.51	0.251	0.51	1	1	1	0.124
Gulf Region	0.51	0.51	0.51	1	0.51	0.51	0.251
Laurentian Regi	1	0.251	0.51	0.51	0.251	0.51	0.124
Central and Arct	0.51	0.51	0.251	0.124	0.251	0.251	0.251
Pacific Region	0.251	0.51	0.251	0.251	1	0.51	0.251

Table 4-1: Local priorities calculation for alternatives in demo model

Objectives:	Sustainable Fisheries & Aquaculture	
Criteria:	Science Excellency	
Regions:\Values:	Absolute Value	Distributive Mode Value
Newfoundland Region	0.251	0.083
Maritimes Region	0.510	0.168
Gulf Region	0.510	0.168
Laurentian Region	1.000	0.330
Central and Arctic Region	0.510	0.168
Pacific Region	0.251	0.083
Total	3.032	1.000

Figure 4-10: Flow chart of assessment calculation of the application system

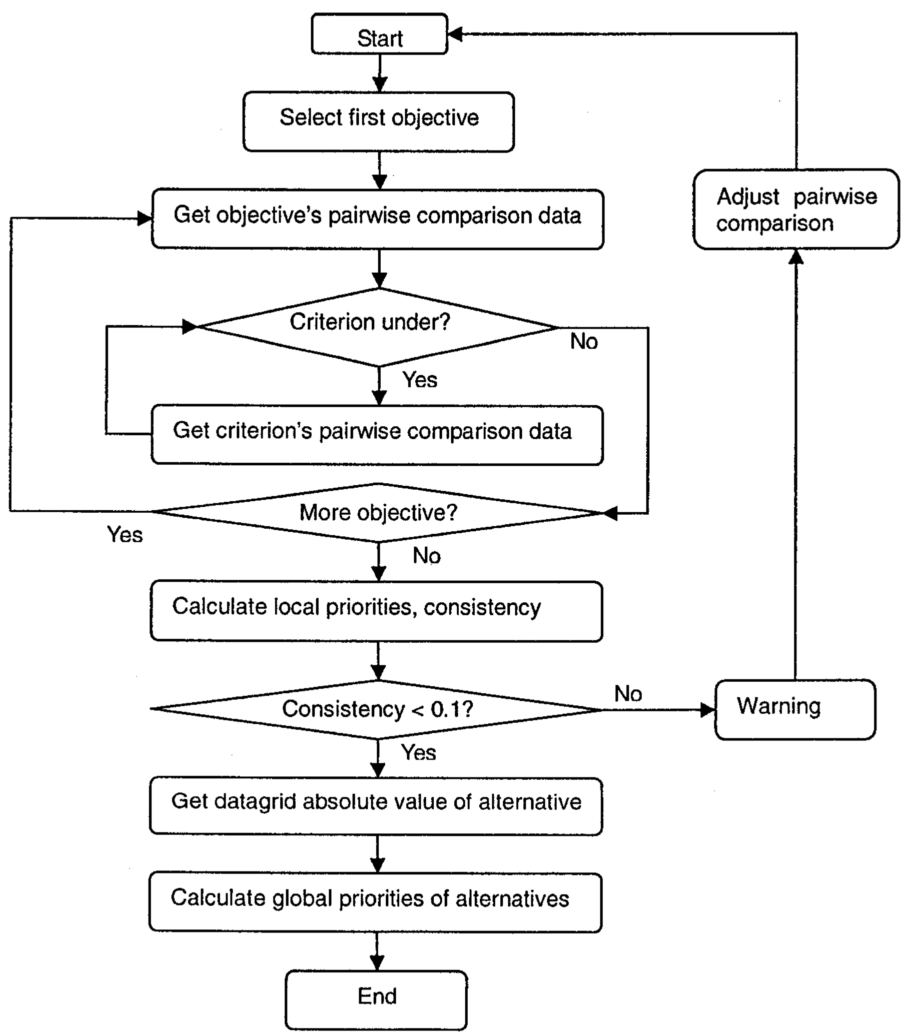


Figure 4-10 shows the calculation procedure for the sample national AHP model. The priorities of objectives and criteria (local priorities) and consistency are gained by calculating the eigenvector and eigenvalue of the pairwise comparison matrix using VB component named Matrix ActiveX Component 3.0 (Appendix VI). The priorities of alternatives (global priorities) are calculated by first formulating the matrix of datagrid absolute value, and then multipling the matrix by the local priorities.

4.1.3 Synthesis and Results

3a. Synthesis report. After all the judgments in the model have been made and priorities have been calculated, a synthesis is automatically performed. The priorities of the elements in the hierarchy are shown following their position in the model (Figure 4-11).

Figure 4-11: Priority of element in hierarchy of system demo sample

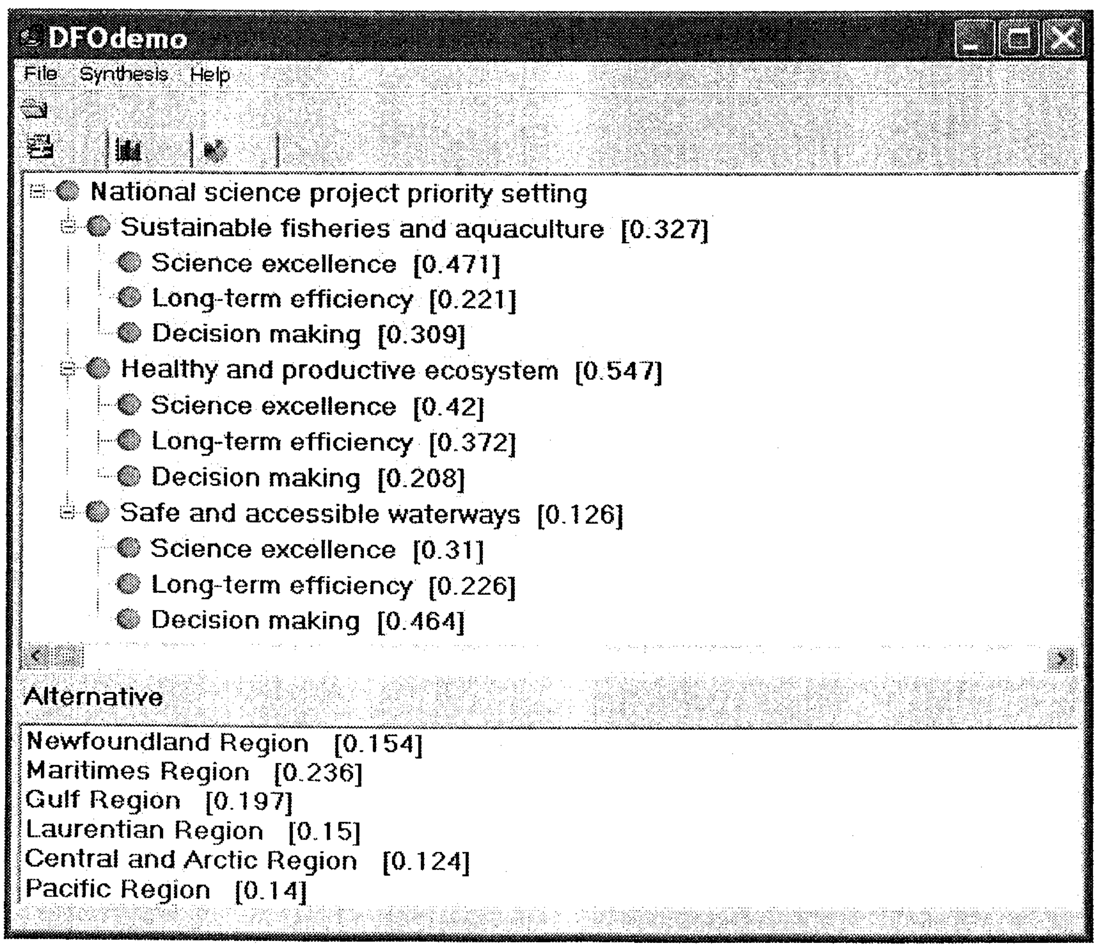


Figure 4-12: Priorities ranking of alternatives of system demo sample

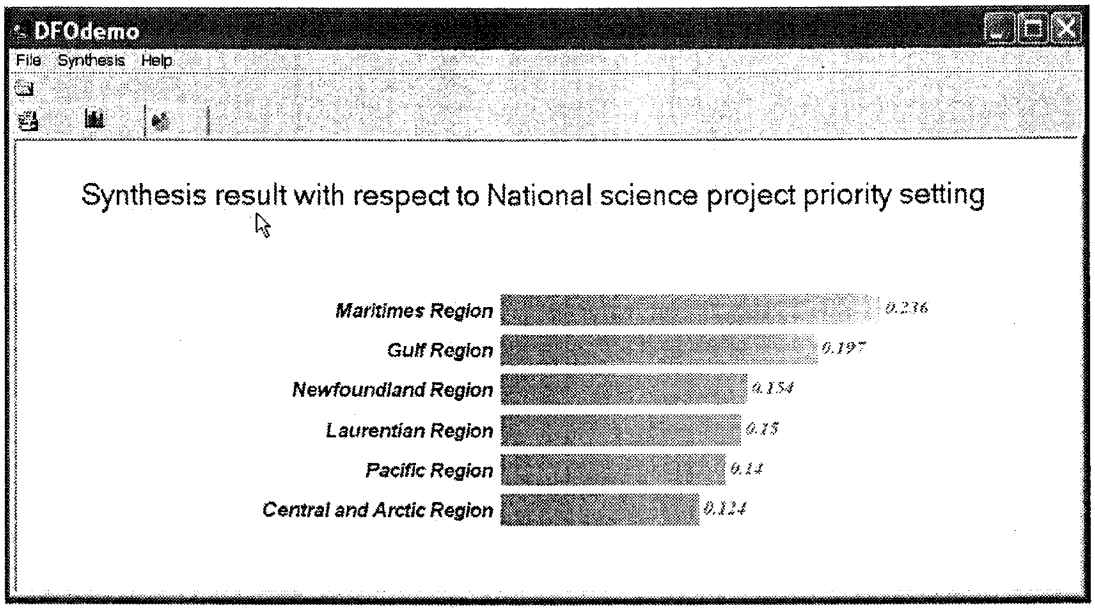
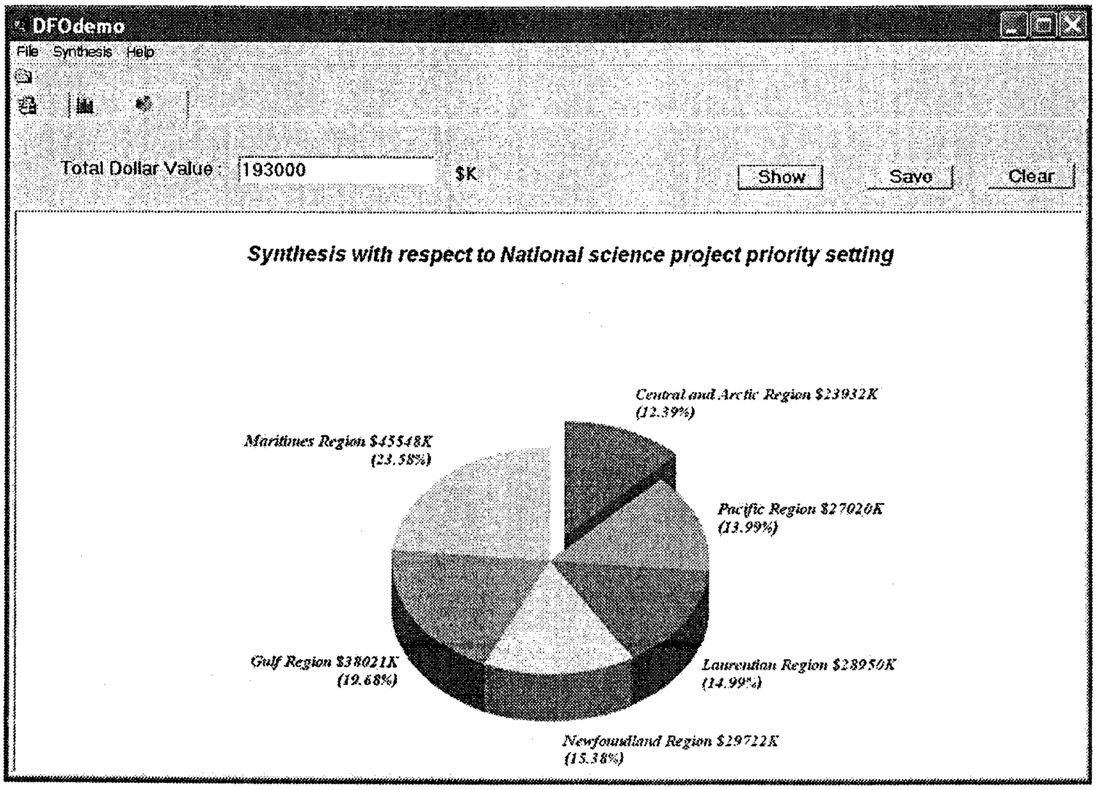


Figure 4-13: Budget allocation of alternatives of system demo sample



The final priorities ranking of alternatives can also be shown in a graphical synthesis report (Figure 4-12). Corresponding to the total budget input by decision maker, the budget allocated to each alternative is illustrated in a pie chart shown in Figure 4-13.

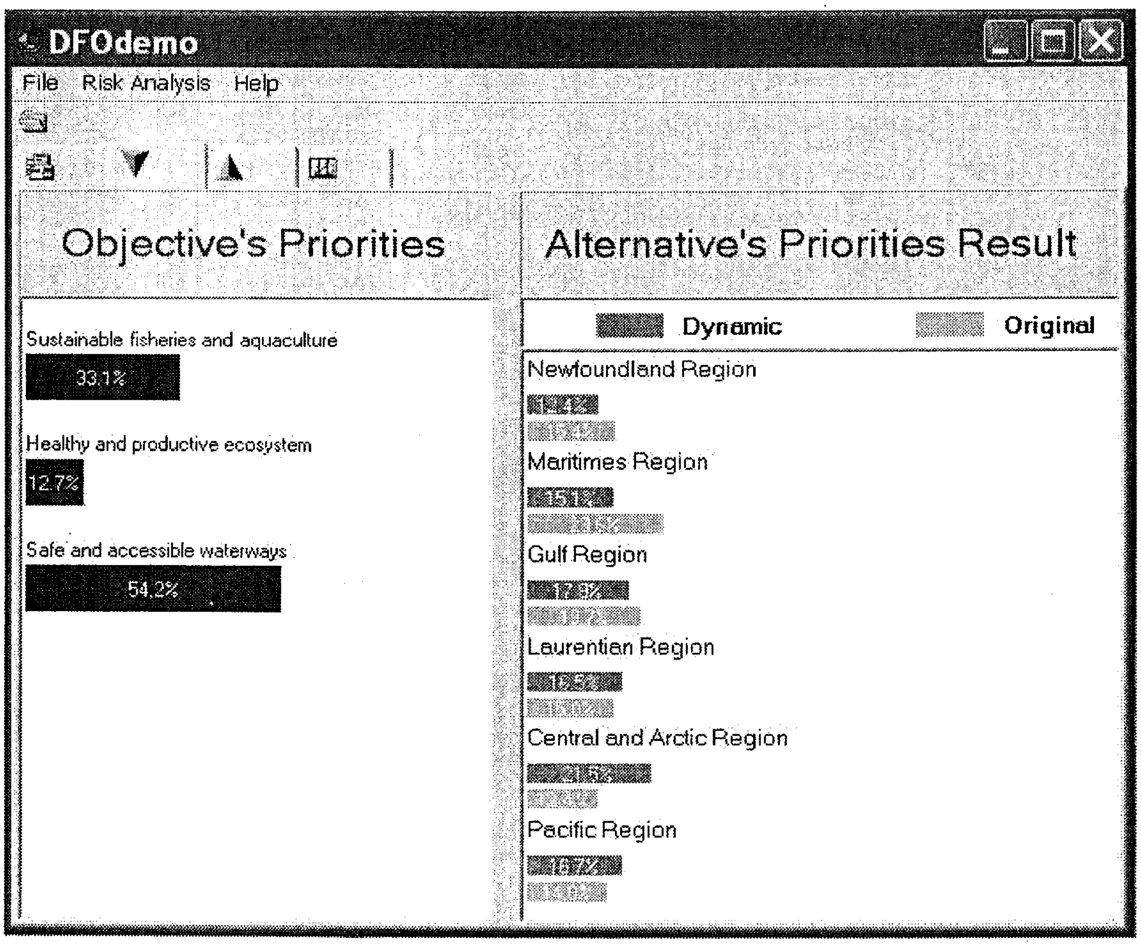
4.1.4 Risk Analysis

3b-3c. Risk analysis and report. Risk analyses can be performed from top-down to show the effect of changes on the objectives of the model, e.g., in Figure 4-14, when the weight of objective "Healthy and productive ecosystem" changes from 0.547 to 0.127, correspondingly, the weight of other objectives and alternatives change. I.e., if the weights of the top level objectives were changed, the priorities ranking of alternatives would change correspondingly, which will show the sensitivity of the alternatives with respect to all the objectives below the goal. These changes shown within the various alternatives responding to the change of different objective are different since the local weights allocated to alternatives toward each objective are different. Through this top-down risk analysis, the influence of individual top level objective on the final decision can be compared according to the corresponding change of the bottom level alternative.

Another way designed in this application system to perform the risk analysis is the bottom-up approach. As described in step 2 "Alternative assessment", instead of using pairwise comparison to decide the relative contribution of each alternative toward every criterion under each objective, in this application, absolute value from the data grid is used to measure the performance of individual alternative with respect to every criterion directly. This bottom-up approach is then designed to show the difference on the resulting priorities ranking of alternatives in the data grid if the performance of one alternative is not the same as what is expected during the decision making process. Since the original decision of budget allocation is made based on the evaluation of alternatives' performance, any difference happen between the evaluated value and actual performances measure of one alternative will cause the percentage of budget allocation shifts among the alternatives. For example, as shown in Figure 4-15, the budget percentage of individual alternative changes when the performance of "Maritimes

Region" has gone up 39.2% (the blue slider) higher than the original evaluation.

Figure 4-14: Top-down objective dynamic of system demo sample



Finally the last sub-module for the risk analysis is designed based on the utility theory introduced in the previous chapter (Figure 4-16). By changing the alternative's threshold value initially calculated using the proposed method and parameters discussed, the utility function of single alternative will change correspondingly (see also the "Utility Curve" in Figure 4-16). Consequently, the total utility over all alternatives, which can be described as the satisfactory level of the current decision, is different. Moreover, from the Total Utility Assessment curve, the total utility varies in response to any change of the "Current Total Budget".

Figure 4-15: Bottom-up alternative performance dynamic of system demo sample

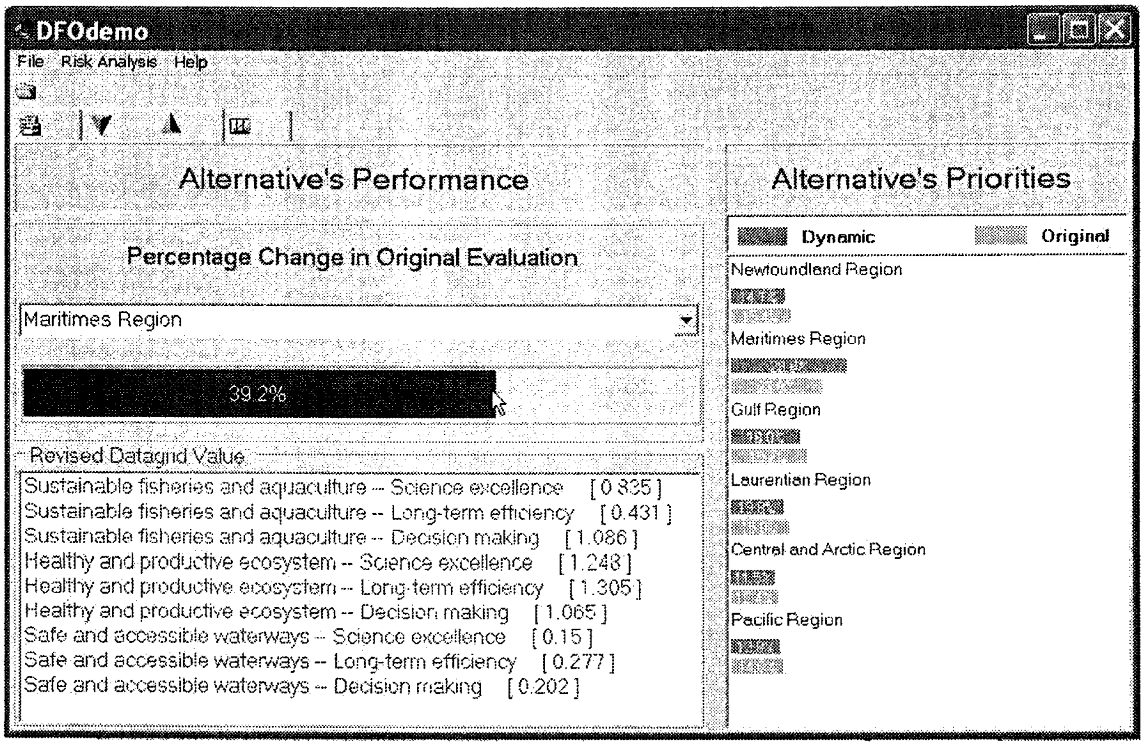
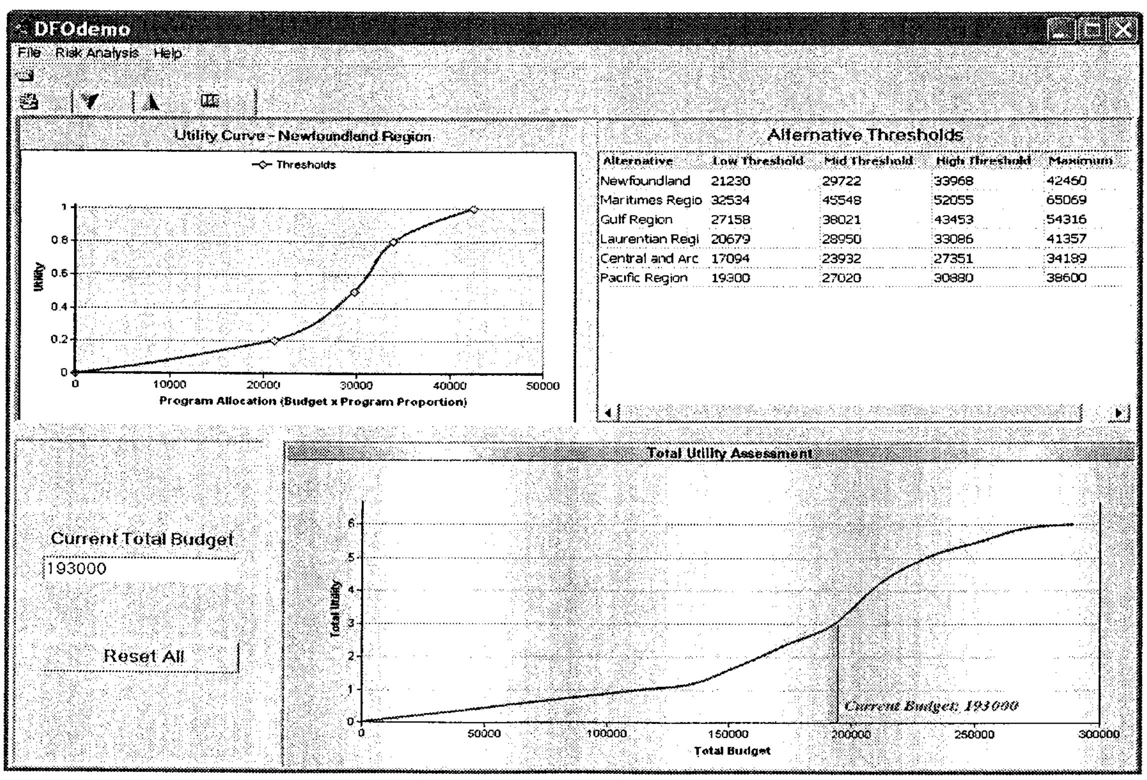


Figure 4-16: Utility analysis of system demo sample



4.2 DFO Application

According to the proposed procedure for the DFO case, the primary step of whole process is to build the model using effective methodology. The discussion of methodology for DFO science prioritization in Chapter 3 demonstrated that there are different problem formulation structures for the DFO case: (1) institutional model formulation (2) functional model formulation (3) multidisciplinary model formulation. The processes for simulating the DFO risk-based science prioritization problem using these formulations are similar with the procedure illustrated in section 4.1. This section combines the various methodologies into actual model building process, and focuses on the linkage and difference within each specific problem formulations during the full simulation process corresponding to the DFO sample.

4.2.1 Institutional Model Formulation

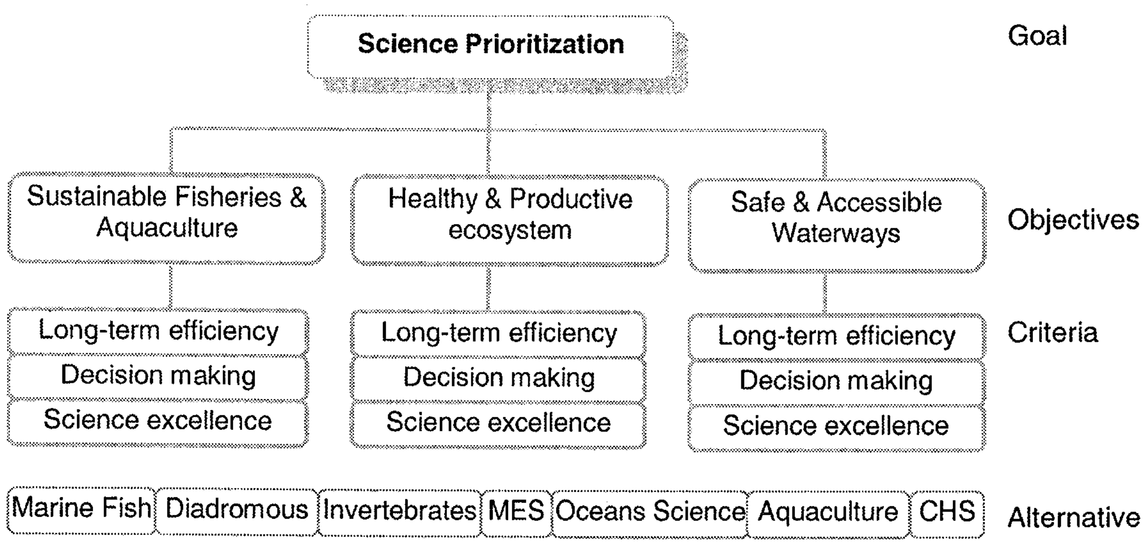
The Institutional model formulation structures DFO's science priority problem applying the principles of allocation from the top down according to the institutional organization. This model requires prioritization take place at each level of the organization with a filtering down of the information ultimately to the lowest Project level.

Each institution within each level of the organization formulates their own hierarchy from goal, objectives to criteria. The institutions within the same level of organization are treated as the alternatives of the upper institution's hierarchy they belong to. In this way, the hierarchies within each level of organization are linked together forming an integrated model. The result prioritization for resource allocation in one level then constrain the next level's budget setting, which will be allocated through the prioritization of next level. Through the prioritizations processing from National down to Project level, the perspective of every institution that can be reflected in their prioritization results then filter down to the final Project.

For example, assuming the hierarchy built for Maritime Region is same as National level model used in section 4.1, and Maritimes Region manages seven

Divisions including Marine Fish, Diadromous Fish, Invertebrates Fisheries, Marine Environmental Science, Oceans Science, Aquaculture, and Canadian Hydrographic Service, then when building the hierarchy for Maritimes Region, these Divisions construction the lowest level of this hierarchy as the alternatives (Figure 4-17).

Figure 4-17: Institutional model sample for Maritimes Region



To build this Regional level model, the system has first to finish the National model building problem. The resulting budget allocations from Institutional model on National level form the total budgets for resource allocation on Regional level including Maritimes Region. Consequently, results of these Regional level prioritizations including the priorities ranking and budget allocation for each Division within the Region are then saved for the decision making process of its next level, the Division level resource allocation plan.

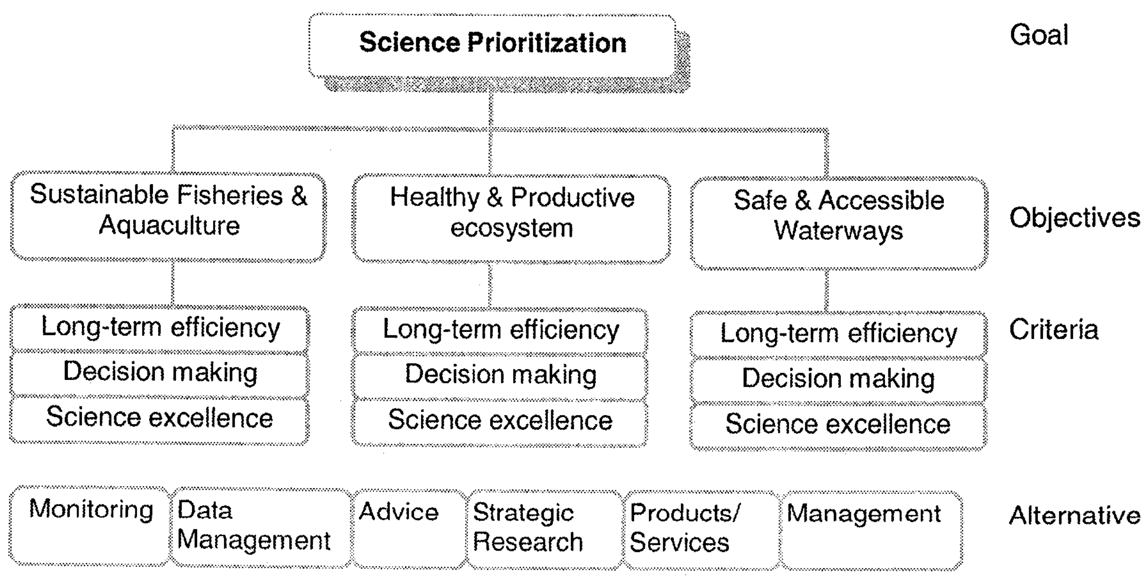
4.2.2 Functional Model Formulation

Functional model formulation structures DFO's science priority problem applying the principles of allocation from the top down according to the DFO Business Line. This model also requires prioritization take place at each level of the organization. However, it assumes homogeneous application of science services described in terms of the DFO Business Line science functions may be applied to the organizational units. The problem remains as to how to allocate these budgets within each level hierarchy.

Still each institution within each level of the organization formulates their own hierarchy from goal, objectives to criteria. However, various functions implemented by the institutions are treated as the alternatives of the independent hierarchy. In this way, the hierarchies within each level of organization are linked together through the homogenous functions. Through the prioritizations processing from National down to Project level, the National perspective DFO science can be filtered down to the final Project in the prioritization results within each level.

For example, assuming the same hierarchy is built for Maritimes Region functional model, and Maritimes Region manages to implement DFO Business Line science functions including Monitoring, Data Management, Advice, Strategic Research, Products/Services and Management, then when building the hierarchy for Maritimes Region, these functions construct the lowest level of this hierarchy as the alternatives.

Figure 4-18: Functional model sample for Maritimes Region



Similar simulation process with the institutional model is used this structured model, only the alternatives of each single hierarchy within the model are different science functions instead of Divisions. Results of National level functional model has been saved for the prioritizations of this model, and accordingly the results of Maritimes Region functional model are then saved for the next level decision making process of the resource allocation plan. The full prioritization process from National level top down to Project level is composed of many basic prioritization processes performing for the hierarchies within each level.

4.2.3 Multidisciplinary Model Formulation

As for the multidisciplinary model formulation discussed in section 3.5, the hierarchy built under this point of view has the same structure as the basic AHP model hierarchy above. It constructs the DFO case bottom up into a single large integrated hierarchical structure. Only in this formulating methodology, regional information has not been treated as a separate level of the hierarchy. On the other hand, it has been synthesized as a property of each project in the lowest level of hierarchy. In this way, the information for DFO prioritization can be integrated into one hierarchy instead of many relative independent hierarchies.

Different with the previous two formulations, the simulation of multidisciplinary model is same with the process of the basic AHP model simulation presented in section 4.1. The prioritization process takes place completely from top down to each project. Since each level within this large hierarchy is built with respect to the perspective from Nation, Division, Program and Project, the resulted priorities of each Project therefore reflect the objectives set for each level.

4.3 DFO Model Comparisons

The comparisons between the results of different model formulation are performed by two components. These components include first the comparison between the resulted allocation of individual model formulation and the actual allocations, and then the comparison between the results from different model formulations.

To evaluate the results of proposed prioritization process in this thesis, the process is then implemented for the institutional model formulation sample and functional model formulation sample described in section 4.2. The pairwise comparison data listed in Appendix I has been chosen to represent the pairwise comparison data from managers for the trade-off on the objectives and criteria in the hierarchy.

A set of datagrid evaluation data is calculated backward based on the data sheet in Appendix V, the aggregation of "Projects Maritimes - by Functions & Tables (Strategic Outcomes) (July 22)" (DFO 2003d). The calculations for determining the datagrid values from these actual data for the Institutional model are explained using the sample in Figure 4-17 following the steps listed below:

1. Obtain the original priorities for each alternative Division, which is its proportion of the grand total (Table 4-2).

Table 4-2: Original data of Divisions in Maritimes project inventory

Divisions in Maritimes Region	Grand Total ($K)	Table 1. Sustainable Fisheries and Aquaculture	Table 2. Healthy and Productive Aquatic Ecosystems	Table 3. Safe and Accessible Waterways	Original Priorities
Marine Fish	6,253	62	38	1	0.149
Diadromous Fish	2,998	21	79	-	0.071
Invertebrates Fisheries	4,536	65	35	-	0.108
Marine Environmental Science	6,518	11	89	-	0.155
Oceans Science	11,515	21	57	21	0.274
Aquaculture	3,496	98	2	-	0.083
Canadian Hydrographic Service	6,767	3	14	83	0.161
Grand Total	42,083	34	47	19	1.000

2. Set the datagrid value of the alternative with the highest original priority for the Criteria under one Objective as the same as the original data shown in the Table 4-2 above. The result is shown in Table 4-3 below.

Table 4-3: Datagrid value for division with highest priority calculated backward using Maritimes project inventory

Objectives:	Sustainable Fish&Aqua			Healthy Ecosystems			Safe Waterways		
Divisions:\Criteria:	SciExc	LT eff	DM	SciExc	LT eff	DM	SciExc	LT eff	DM
Oceans Science	0.212	0.212	0.212	0.574	0.574	0.574	0.214	0.214	0.214

3. Repeat step 2 for other alternatives, then reduce the values by the scale of their Original Priorities versus the highest one, e.g., datagrid value of Marine Fish is shown in Table 4-4.

The resulting allocation of this sample Institutional model is shown to be close to the actual allocations with minor differences (Table 4-5).

Table 4-4: Datagrid value for Marine Fish Division calculated backward using Maritimes project inventory

Objectives:	Sustainable Fish&Aqua			Healthy Ecosystems			Safe Waterways		
Divisions:\Criteria:	SciExc	LT eff	DM	SciExc	LT eff	DM	SciExc	LT eff	DM
Marine Fish	0.343	0.343	0.343	0.209	0.209	0.209	0.003	0.003	0.003

Table 4-5: Calculated allocation for Institutional model comparing with original priorities from Maritimes project inventory

Divisions	Original Priorities	Original Allocations	Resulted Priorities	Resulted Allocations	Difference
Marine Fish	0.149	6,253	0.156	6,560	5%
Diadromous Fish	0.071	2,998	0.078	3,282	9%
Invertebrates Fisheries	0.108	4,536	0.114	4,792	6%
Marine Environmental Science	0.155	6,518	0.183	7,682	18%
Oceans Science	0.274	11,515	0.274	11,515	0%
Aquaculture	0.083	3,496	0.078	3,268	-7%
Canadian Hydrographic Service	0.161	6,767	0.118	4,978	-26%

As for the sample functional model shown in Figure 4-18, the backward calculation is explained by the following steps:

1. Transfer the percentage of Divisions' contribution toward science functions into dollar values; sum these values for each function; divide the sum by the total value to obtain the original priorities for each function (Table 4-6).

2. Use the aggregated contribution percentages as the contribution of every function toward each objective and set the datagrid value of alternative with the highest original priority toward the criteria under each objective as

discussed for the institutional model as Table 4-7 below.

Table 4-6: Original data of Maritimes functions calculated backward using Maritimes project inventory

Divisions in Maritimes Region	Grand Total ($K)	Monitoring / Surveillance	Data Management / Gestion de données	Advice / Conseil	Strategic Research / Recherche stratégique	Products/Services / Produits/Services	Management / Gestion	Table 1. Sustainable Fisheries and Aquaculture	Table 2. Healthy and Productive Aquatic Ecosystems	Table 3. Safe and Accessible Waterways
Marine Fish	6,253	1,740	117	2,015	1,764	-	617	62	38	1
Diadromous Fish	2,998	481	32	297	874	1,007	307	22	78	
Invertebrates Fisheries	4,536	704	237	1,518	1,350	-	728	47	54	
Environmental Science	6,518	265	192	843	5,022	18	179	15	85	
Oceans Science	11,515	2,785	499	1,462	4,713	763	1,293	23	54	23
Aquaculture Science	3,496	-	-	38	3,191	-	267	99	1	
Hydrography	6,767	2,479	884	140	826	2,113	325	6	13	81
Grand Total	42,082	8,453	1,961	6,312	17,740	3,901	3,715	34	47	19
Original Priorities	1	0.20	0.05	0.15	0.42	0.09	0.09	34	47	19

3. Repeat step 2 for other functions, and then reduce the values by the scale of their Original Priorities versus the highest one.

Table 4-7: Datagrid value for function with highest priority calculated backward using Maritimes project inventory

Objectives:	Sustainable Fish&Aqua			Healthy Ecosystems			Safe Waterways		
Function:\Criteria:	SciExc	LT eff	DM	SciExc	LT eff	DM	SciExc	LT eff	DM
Strategic Research	0.340	0.340	0.340	0.468	0.468	0.468	0.193	0.193	0.193

The resulting allocation of prioritization process is same as the actual allocations as shown in Table 4-8 because the contribution percentage toward each objective of every functions is same, which makes the scales of each resulted priorities versus the highest priority among the functions match the scales calculated by the original priorities exact.

Table 4-8: Resulted allocation for Functional model comparing with original priorities from Maritimes project inventory

Functions	Original Priorities	Original Allocations	Resulted Priorities	Resulted Allocations	Difference
Monitoring	0.201	8,453	0.201	8,456	0%
Data Management	0.047	1,961	0.047	1,961	0%
Advice	0.150	6,312	0.150	6,314	0%
Strategic Research	0.422	17,740	0.422	17,746	0%
Products/Services	0.093	3,901	0.093	3,902	0%
Management	0.088	3,715	0.088	3,716	0%

The comparative results illustrated above show that the proposed prioritization process can accurately reflect the actual situation in the Maritimes Region budget allocation. Furthermore, to compare the results between different model formulations, the resulting allocation of the Maritimes Region functional model is recalculated back into the budget allocation of Divisions by following steps:

1. Obtain the proportion of dollar values allocated to every function in each Division to the total budget as shown in Table 4-9.

Table 4-9: Original proportion by function / Division from Maritimes project inventory

Divisions	Grand Total ($K)	Monitoring / Surveillance	Data Management / Gestion de données	Advice / Conseil	Strategic Research /Recherche stratégique	Products/Services / Produits/Services	Management / Gestion
Marine Fish	6,253	0.041	0.003	0.048	0.042	0.000	0.015
Diadromous Fish	2,998	0.011	0.001	0.007	0.021	0.024	0.007
Invertebrates Fisheries	4,536	0.017	0.006	0.036	0.032	0.000	0.017
Environmental Science	6,518	0.006	0.005	0.020	0.119	0.000	0.004
Oceans Science	11,515	0.066	0.012	0.035	0.112	0.018	0.031
Aquaculture Science	3,496	0.000	0.000	0.001	0.076	0.000	0.006
Hydrography	6,767	0.059	0.021	0.003	0.020	0.050	0.008
Grand Total	42,082	0.201	0.047	0.150	0.422	0.093	0.088
Original Priorities	1	0.201	0.047	0.150	0.422	0.093	0.088

2. Recalculate the proportions according to the resulting priorities of functions. I.e.

$$new\ proportion = \frac{old\ proportion}{original\ priority} \times resulted\ priority.$$

3. Recalculate the dollar values based on the new proportions and accordingly the allocation to each Division shown in Figure 4-19.

The comparison between this allocation recalculated from the Maritimes Region functional model and the one resulting from the institutional model (Figure 4-20) shows that the allocations resulting from both model formulations are close to each other and also that of the actual allocation found in the Maritimes Region project inventory (DFO 2003d).

Figure 4-19: Division's allocation from Maritimes Region Functional model

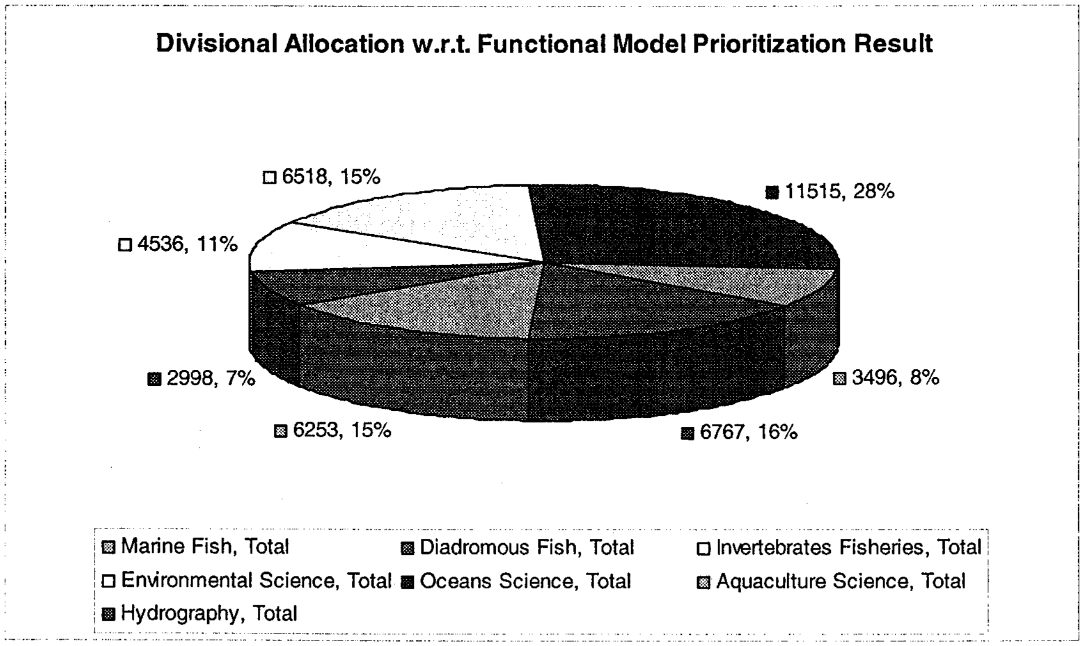
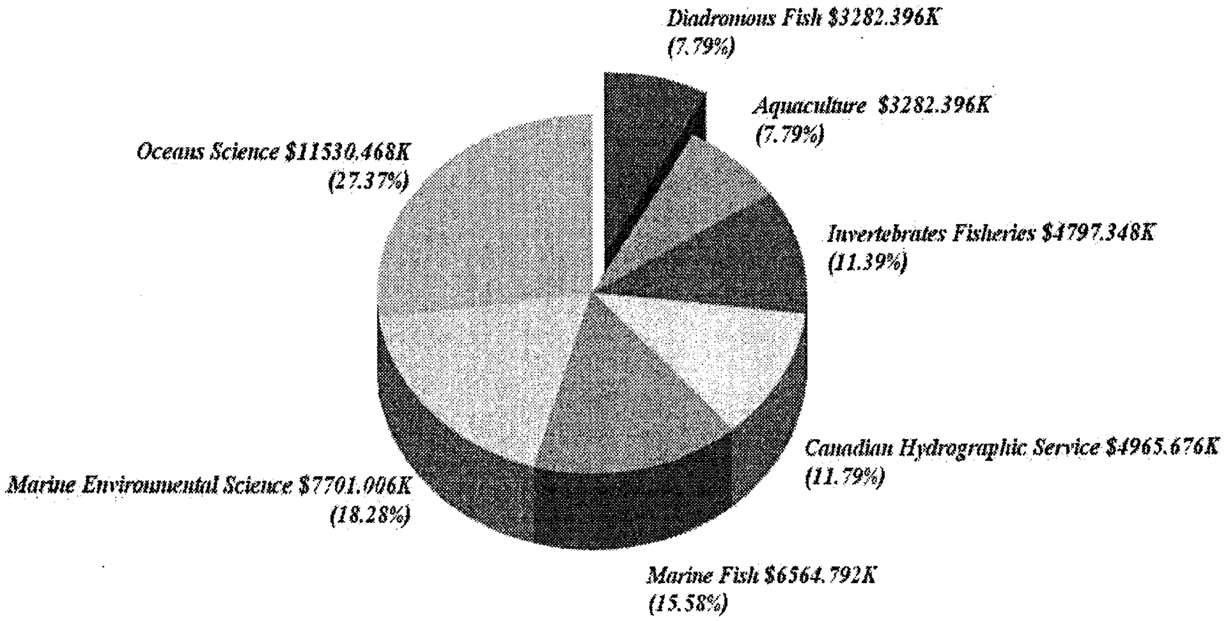


Figure 4-20: Divisional allocations of Maritimes Region Institutional model



5. Case Study

In this chapter, the application of risk-based science priorities setting system is described for an experiment carried out at the DFO Science Maritimes Region Bedford Institute of Oceanography (BIO). This experiment was part of an invited workshop where risk-based prioritization methods were tested and applied to specific cased of direct interest to DFO science Maritimes Region. The workshop was arranged by the director and managers in DFO Science Maritimes Region as a two-day workshop to carried out the problem and experiment that was developed as part of the research on a DFO Maritimes Region related problem.

There are 30 participants including the director and managers/decision maker in DFO Science Maritimes Region, and members of management community of the BIO science institution in the Region. During the workshop, the thesis research was presented and feedback was collected the group discussion and the management meeting of the directors at the next day.

First, background information, current problems and issues, and some general outlines for the BIO risk-based prioritization exercises are provided. Next, the proposed priority setting process is applied and examined in the context of the specific case of the Maritimes Region science monitoring program data. Finally, the results are analyzed and the discussion on risk issues specific to DFO science is elaborated for providing direction for further system development.

5.1 Introduction

Government's resource reallocation has been directed to transform old expenditure to new or higher priorities according to the September 2002 Speech from the Throne. This direction is set for all departments including DFO. In May 2003, DFO was labeled as one of the departments participating in a Treasury Board Expenditure and Management Review (EMR) of the year, which is a review for department's mandates and programs announced by the President of the Treasury Board to ensure expenditures continue to contribute relevant results for Canadians. Responding to the diverse challenges of maintaining or improving overall operational effectiveness with limited resources, DFO has begun a comprehensive Departmental Assessment and Alignment Project (DAAP) since November 2002 to assess its policies, programs and activities. (DFO 2003c, g)

The purpose of this review is to enhance DFO's financial stability for the future, strengthen the current management system and policies, and reassess its programs. The overarching objective of the DAAP is "to strengthen the Department's ability to work with Canadians to ensure the sustainable development and safe use of Canadian waters." The DAAP has therefore been guided by principles that support sustainable development: integrated resource management, partnerships, the concept of the public good, and risk management and precaution. (DFO 2003g)

The DAAP is working to fully assess all of DFO's expenditures, identify ways to work more efficiently and modernize management and financial planning processes. In line with the DAAP options for securing financial stability to support DFO programs and services, the emphasis will be on identifying and improving program efficiencies. This makes DFO remain committed to performance measurement. Consequently the development and introduction of a new comprehensive performance measurement system that supports decision making and ensures accountability is considered one of the outcome following the DAAP report. (DFO 2003g)

The DAAP has set the stage for the renewal of DFO's Strategic Plan that provides directions for all individual science programs. However, the department still needs a unifying and integrating decision making framework to pursuit the program reassessment and resources reallocation. This decision making framework should be linked to evaluation methods and program or project performance indicators to assist in ranking operational work priorities. The pre-evaluation of proposed projects' performance relative to the DFO science strategic goals is uncertain. Estimating expectations of projects' performance requires the development of a risk-based approach to ranking projects in the planning framework.

Methods for evaluating risk-based priority-based decision alternatives can be seen in the wider context of constrained optimization problems. More specifically, these problems can be formulated as constrained multicriteria decision making problems where decisions must be taken using scare resources to respond to different and competing program needs to reach specific objectives. The proposed risk-based science prioritization process introduces an applied approach to evaluate projects and to determine their expected contribution to the organization's goals from their anticipated performance results.

In this case study, the risk-based science prioritization using multicriteria approach has been demonstrated using DFO Science Maritimes Region institutional and functional sample models. Following the presentation, the application system has been further applied to DFO Science Maritimes Region monitoring programs and programs identified as contributing to Healthy and Productive Aquatic Ecosystems. The application of the AHP multicriteria methodology will take place as data input exercise by five breakout groups set up for the monitoring programs priorities setting exercise. The application process and results will be summarized and discussed in the following section. The application of political and conflict case are not included at the regional level of science prioritization. These are assumed to be absorbed in the formal prioritization process for the region. Each Breakout Group's feedback datasheet are provided in Appendix III along with a summary graphic report of these Group results. Aggregate results summarizing all Breakout Groups' feedback are also provided as an average evaluation for each cell of the datasheet.

5.2 Maritimes Science Project Inventory

This application process presented in the BIO workshop is built upon the gracious assistance from the Maritimes Region Science Branch personel by providing the detailed description of the region's ongoing projects. Projects information is used as the data source for the application system and enlisted into a comprehensive document entitled "Science Projects Inventory - Maritimes Region Science Branch, FY 2002-03" (DFO 2002) and "Projects Maritimes - by Functions & Tables (Strategic Outcomes) (July 22)" (DFO 2003d), a spreadsheet file in Excel format identifying some 400 regional science programs and their expected contributions to DFO objectives and functions.

Basically, the projects in the Maritimes Region project inventory documentation has been organized following DFO science sector fourth-level Business Lines. The definition of 2^{nd}, 3^{rd} and 4^{th} level Business Lines is included in Appendix II. Based on the Business Line organization, these two documents categorize all the project information further following different arrangements.

The first document compiles the detail information of individual science project in Maritimes Region by responsibility center and project code. The contents of this document depict individual science project by its Project Title, Project Leader, Section, Division, Location, Description, Expected Results, 4^{th} level Business lines, money figure and etc. It has been a significant assistance in understanding the content in the second document, which contain the main data source for the sample application demonstrated in the workshop.

The spreadsheet "Projects Maritimes - by Functions & Tables (Strategic Outcomes) (July 22)" organizes the Maritimes Region science projects by DFO strategic outcomes (Tables) discussed before and also by six science functions including: (1) Monitoring; (2) Data Management; (3) Advice; (4) Strategic Research; (5) Products/Services; (6) Management. The data sheet characterizes individual science project numerically by different dollar values assigned to it and the percentage of its contribution toward every Function and Table. During the prioritization process, the data in this spreadsheet has been aggregated for different model construction purposes.

As for the Maritimes Region institutional model, different Divisions form the lowest level of hierarchy, alternatives. Accordingly, the project inventory has been aggregated based on the composition relationship between 4^{th} level Business Lines and Divisions shown in Table 5-1. The dollar value in "Grand Total" was chosen as the budget value in the modeling process. Then budget allocation to each Division can be calculated by simply summing up the "Grand Total" of all projects under the Division, moreover, by dividing the allocation by Region's

"Grand Total", the proportion of Divisions' allocations were obtained and treated

Table 5-1: Institutional aggregation basis for Maritime project inventory

Fourth-Level Business Line	Division
Fisheries Research -111	1. Diadromous Fish 2. Invertebrates Fisheries 3. Marine Fish
Fisheries Research, General - 1110	Marine Fish
Diadromous and Freshwater Fish - 1111	Diadromous Fish
Groundfish - 1112	Marine Fish
Pelagic Fish - 1113	Marine Fish
Invertebrates and Other Species - 1114	Invertebrates Fisheries
Marine Mammals - 1115	Marine Fish
Marine Plants - 1116	Invertebrates Fisheries
Environmental Science - 112	Marine Environmental Science
Environmental Science, General - 1120	Marine Environmental Science
Toxic Chemicals - 1121	Marine Environmental Science
Fish Habitat - 1123	Marine Environmental Science
Environmental Effects of Aquaculture - 1124	Marine Environmental Science
Environmental Data & Advice - 1125	Marine Environmental Science
Biodiversity Science - 113	1. Diadromous Fish 2. Invertebrates Fisheries 3. Marine Fish
Biodiversity Science - 1131	Invertebrates Fisheries
Species at Risk - 1132	1. Diadromous Fish 2. Invertebrates Fisheries 3. Marine Fish
Oceans Science - 121	Oceans Science
Oceans Science, General - 1210	Oceans Science
Phys. & Chem. Oceanogr., Living Res. - 1211	Oceans Science
Biological Oceanography, Ecology - 1212	Oceans Science
Env. Interactions with Marine Activity - 1213	Oceans Science
Ocean Climate - 1214	Oceans Science
Ocean Data, Inf & Advisory Services - 1215	Oceans Science
Aquaculture Science - 122	Aquaculture
Aquaculture Science, General - 1220	Aquaculture
Invertebrate Aquaculture - 1222	Aquaculture
Finfish Aquaculture - 1223	Aquaculture
Aquaculture-Environment Interactions - 1225	Aquaculture
Hydrography - 131	Canadian Hydrographic Service
Hydrography, General - 1310	Canadian Hydrographic Service
Hydrographic Surveys - 1311	Canadian Hydrographic Service
Hydrographic Products - 1312	Canadian Hydrographic Service
Technology Applications & Transfers - 1314	Canadian Hydrographic Service
Program Mgnt and Technical Support - 151	1. Regional Science Director's Office 2. Aquaculture
Executive and Regional Science Management - 1510	Regional Science Director's Office
Management & Support, General - 1511	1. Regional Science Director's Office 2. Aquaculture
Technical Support - 1512	Regional Science Director's Office
Vessel Support - 1513	Regional Science Director's Office

as the Divisions' priorities. To aggregate the Division's contribution toward the three strategic outcomes, first transform the percentage value of individual project to dollar value, and then use the same calculation to get the aggregated proportion of Division's contributions. These proportions were treated as the sum of the absolute values of Division's contribution toward criteria. Finally, by proportionally assign these absolute values into each criterion, the datagrid value for Maritimes Region institutional model was created as part of the input data. The same backward calculation process was implemented to prepare the data for the Division and Program level modeling.

The aggregation process for the Maritimes Region functional model is similar with the process discussed above; only the data were aggregated by six science functions that compose the alternatives of model. The percentage value of project contribution toward each function was first transformed into dollar value. Next, the same backward calculation process was performed to for the datagrid input. Also, the same process was used for the Division and Program level models.

5.3 Maritimes Monitoring Program Prioritization

The strategic outcomes of DFO science have rapidly evolved and become critical in terms of ensuring the sustainable development of Canada's marine and freshwater ecosystems. With response to the trend and pressure for the renewal of the policies and programs under this strategic plan, which is dependent on a sound science foundation, DFO science disciplines need to work collectively toward more comprehensive assessments of aquatic ecosystems to provide effective monitoring data, which plays an important role in providing essential advice to ensure that efforts are focused on fulfilling key DFO responsibilities.

Monitoring programs carried out by DFO science focus on collecting basic core data needed to monitor the status of the marine ecosystem. Regarding DFO's ecosystem monitoring activities from DAAP science assessment, these monitoring programs are expected to be delivered at a lower cost and more successful implementation. The objective for monitoring programs requires proper program priorities to provide effective and affordable resource allocation. The priorities setting exercise conducted for Maritimes monitoring programs provides an approach to set up the program priorities serves for this purpose.

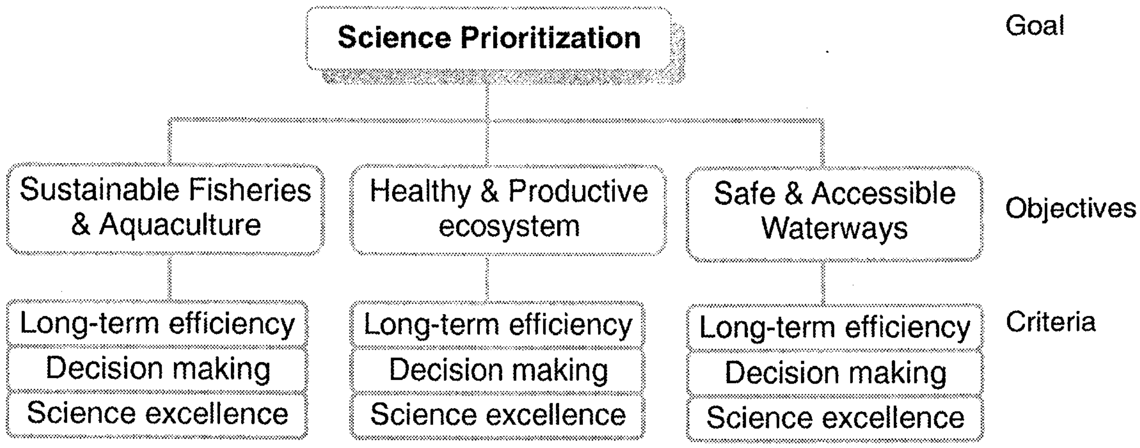
5.3.1 Model Building

The hierarchy used for Maritimes monitoring program priorities setting is originated from National level hierarchy discussed in previous chapter. It embodies the current national perspectives and policy thrusts of the department. Most recently, they were presented in the context of the DAAP and were more directly stated as "Program Outcomes" that followed directly from the stated single strategic objective of "sustainable development and safe use of Canadian waters" in the summary of "The DAAP Final Report" submitted to workshop participants (DFO 2003c), p.7. As well, in the Oceanstone presentation by HQ Science Policy (February 2003), the same 3 objectives were stated as "3 proposed strategic outcomes".

The risk areas explicitly identified on "Risk-based Approach to Priority Setting" (DFO 2003b) also presented at the Oceanstone Maritimes Region Science Retreat by HQ Science Policy are considered the multiple criteria/attributes. These risk areas included the three strategic objectives referred to above, along with 4 items as follows:

- Scientific excellence and credibility
- Long-term efficiency
- Sound and relevant decision-making
- Meeting obligations

Figure 5-1: Maritimes monitoring program hierarchy



The last risk area referred to above, "fulfilling obligations" is interpreted as a "catch-all" attribute that substitutes for absolute program/project performance. As such, de facto, the importance of fulfilling all obligations of all programs/projects and objectives is unchanging. It is not therefore appropriate to speak of the relative importance of fulfilling obligations for a program/project or an objective since the nature of an "obligation" is always that it must be achieved. For this reason, this attribute is not included in the problem formulation (Figure 5-1). The evaluation of different programs'/projects' performance is then applied across the remaining three attributes of (1) scientific excellence, (2) long-term efficiency, and (3) sound decision making. Programs/projects with high relative performance on these multiple attributes describe programs/projects that perform well, and achieve their objectives to a greater extent while meeting their obligations.

Table 5-2: Aggregated monitoring programs

Aggregate Monitoring Program	Description
1. Fisheries Research Monitoring	Includes International Observer Program, National Sampling Program, Oceanographic Data Handling
2. Diadromous Fish Monitoring	Includes DFO-NSPC Partnering Agreement, IRF IBOF 3-Rivers Monitoring, Mactaquac Fish Collection Facility, Maritimes Non-Salmon, Nova Scotia and Inner Bay of Fundy Salmon, South and Western N.B. Salmon
3. Groundfish Monitoring	Includes Groundfish Trawl Surveys, Sentinel Survey - Scotian Shelf
4. Crustacean and Invertebrates	Includes Crustacean Sampling Databases, Full Bay Scallop Association FY 02-03, Offshore Clam Research Project, Research Coordination/Gulf of Maine Crustacean Fisheries Section, Sea Cucumber 4X/5Z
5. Toxic Chemical Monitoring	Includes Clay-Oil Flocculation Monitoring, Point Lepreau Environmental Monitoring Program
6. Ocean Science Monitoring	Includes Continuous Plankton Recorder (CPR) Sampling Program, Section Overhead, Shelf Climate Variability
7. Marine Ecosystem Productivity Total	Includes three Marine Ecosystem Productivity projects
8. Env. Interact's with Marine Activity	Includes Brooke Ocean Technology, Coastal Habitat Studies
9. Climate Change	Includes Climate Change Impact on Energy Sector, Coastal Ocean Climate
10. Hydrographic Monitoring	Includes Hydrography General, Labrador Survey - PI 2, Multibeam Seafloor Mapping, Terra Remote Sensing Inc.

The alternatives chosen for Maritimes monitoring programs priorities setting process have been selected from "Projects Maritimes - by Functions & Tables (Strategic Outcomes) (July 22)" (DFO 2003d). To simplify the calculation for priorities setting process, first, the projects in the Project Inventory with lower than 50% contribution toward function "Monitoring" have been eliminated; then, the remained projects were aggregated in the way shown in Table 5-2 to further reduce the number of alternatives. The corresponding data of the aggregated monitoring programs including "Grand Total", contribution proportion toward each function and strategic outcome were then calculated by the aggregation rules explained in section 5.2.

5.3.2 Evaluation and Data Collection

Based on the problem formulation above, pairwise comparison data for elements within the hierarchy was chosen representing aggregate feedback from a group of six Maritimes Region managers compiled in 2003 following the Oceanstone meeting.

To perform the evaluation on alternatives, 29 individual scientists from different Divisions within the region were invited and properly divided into five Breakout Groups. Each breakout group was assigned to a separated office to complete an evaluation of the 10 Monitoring Programs on an importance grid with respect to the contribution of each program to the science objectives: (1) Sustainable Fisheries and Aquaculture; (2) Healthy and Productive Aquatic Ecosystems; and (3) Safe and Accessible Waterways and the objectives' criteria of Scientific Excellence, Long-term Efficiency, and Sound Decision-making. Discrete measures of 1 to 5 were used, where 1 denoted greatest level of importance and 5 denoted least level of importance of each program in contribution to each combination of objectives and criteria (Table 5-3).

Table 5-3: Program Absolute Evaluation – Scores Table

Program Intensity Description	Score:
Great	1
Significant	2
Moderate	3
Little	4
Very Little	5

As discussed in previous chapters, participants with different expertise and concerns will show different interest during the evaluation process. Correspondingly these differences will have explicit influence on the resulted priorities ranking. Hence, the discussions within each Breakout Group are important for the following result analysis.

The members of Breakout Group #1 have different opinions on defining 'Long-Term Efficiency' criteria and also specific projects within aggregated monitoring program. Furthermore, to better evaluate the importance toward every objective/criteria for each programs, it is crucial on setting the current or desirable perspective and defining the element within the hierarchy. As for the method of data collection, direct data entry using row wise (program) view is preferred by this group for simplification.

The same argument concerning the definition of criteria has risen in Breakout Group #2 when the group members proceed the data entry by row and column. Based on the discussion on individual cells within datagrid, consistency is believed lacking in problem framework definitions. During the evaluation process, the attempts of not overuse values at high and low level were emphasized, and solved by setting at one level and then adjust it according to the difference among programs.

Column-wise approach was chosen by members in Breakout Group #3 in order to allow relative comparison along programs. Importance acquired across the whole programs by setting level. There are also certain comments prompted by "my interpretation is ..." on measuring of different terms e.g., "science excellence", because of the difficulties on differentiating the programs contribution under each criterion/objective, which include the high correlation between the objectives "Healthy and Productive Ecosystems" and "Sustainable Fisheries & Aquaculture".

As for Breakout Group #4, the perspective for deciding the importance was based on current view on programs contributions. Group members debated on defining implication of elements of aggregated programs, e.g. climate change versus oceans science, and terms, e.g. science excellence. Details of individual program including name, FTE, and Grand Total were identified through aggregated monitoring program file. Column wise (object by criteria) approach was used for direct data entry.

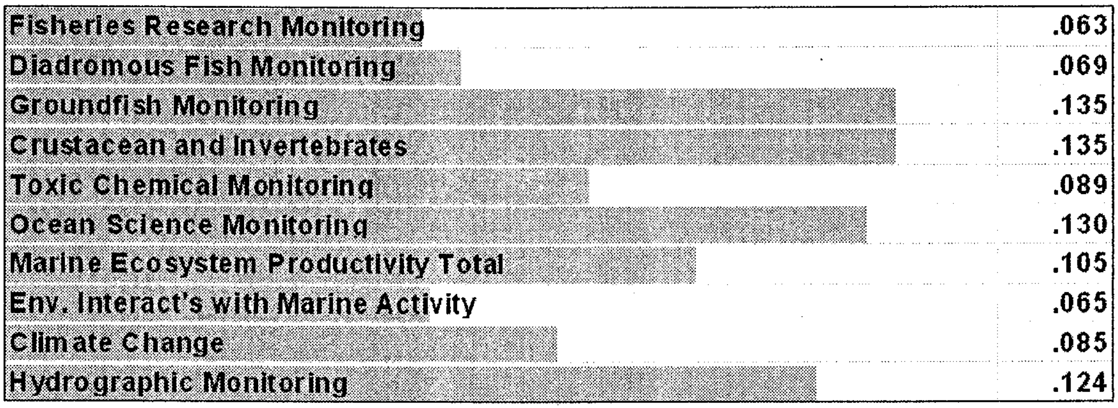
People from Breakout Group #5 generally focus on Hydrographic Monitoring as "ideal" program, which can be seen from the feedback datasheet (Appendix III) and the result analysis for monitoring program priorities setting in the next section.

5.3.3 Summary Results

The result of each Breakout Group's input evaluation was analyzed using the Analytical Hierarchy Process through the Expert Choice software package (www.expertchoice.com). The results shown in Appendix III for each Breakout group as well as the Aggregate results are taken directly from the summary output report from Expert Choice. These results define the weighted hierarchy for the application of the science objectives and criteria. The hierarchy assumed is the same for all Breakout groups and represents aggregate feedback from a group of six Maritimes Region managers compiled in 2003 following the Oceanstone meeting. The results discussed briefly below show the scaled "allocation" made to each Monitoring Program as a consequence of the information compiled from each Break Groups' importance grid feedback.

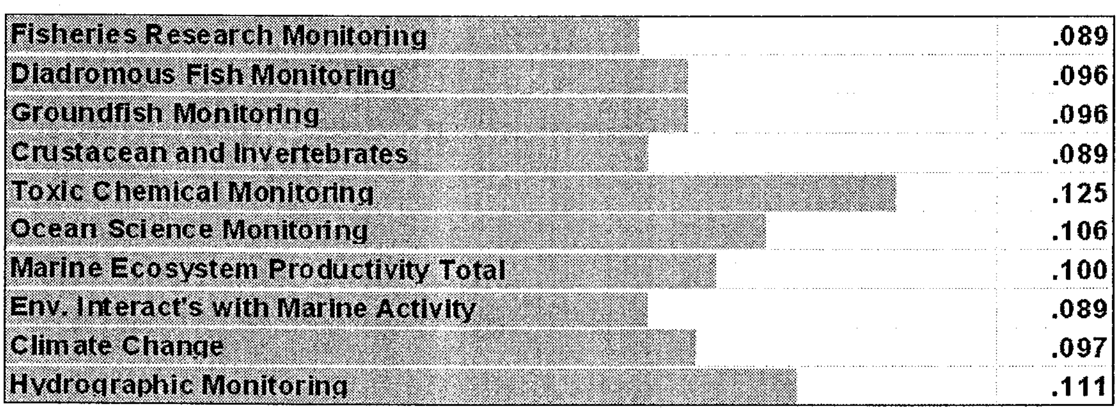
Breakout Group #1 (Figure 5-2). This group favors two programs including (1) Groundfish Monitoring and (2) Crustacean and Invertebrates, which are calculated as having the same top priority. Ocean Science Monitoring and the following program, Hydrographic Monitoring, are programs with the next highest priorities only slightly below those of the two highest priority ranking programs. Other programs fall significantly behind these top ranked programs. Fisheries Research Monitoring has the lowest priority of all the programs in the resultant priorities ranking.

Figure 5-2: Maritimes monitoring program priorities ranking for group #1



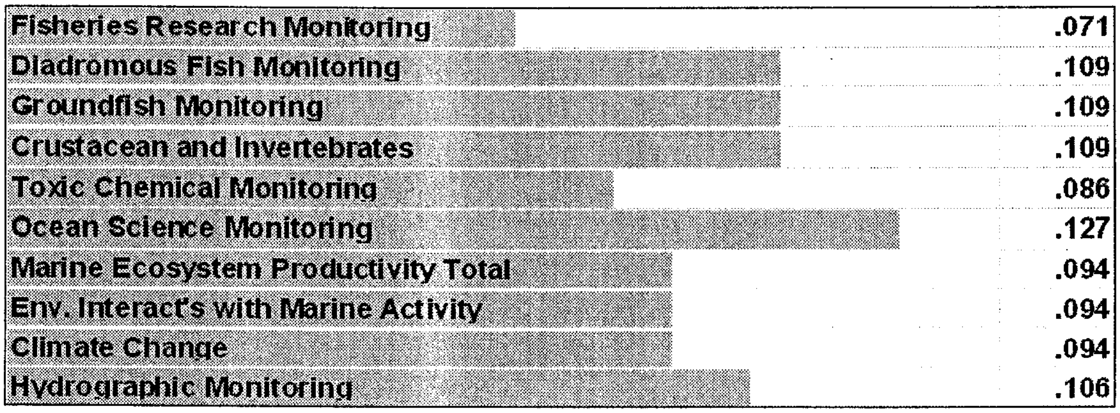
Breakout Group #2 (Figure 5-3). This group evaluates Toxic Chemical Monitoring with the top priority ranking followed closely by Hydrographic Monitoring. The closeness of the input evaluations results in priorities of other programs that are not much different. Fisheries Research Monitoring again ranks as the lowest priority for Group #2 as for Group #1.

Figure 5-3: Maritimes monitoring program priorities ranking for group #2



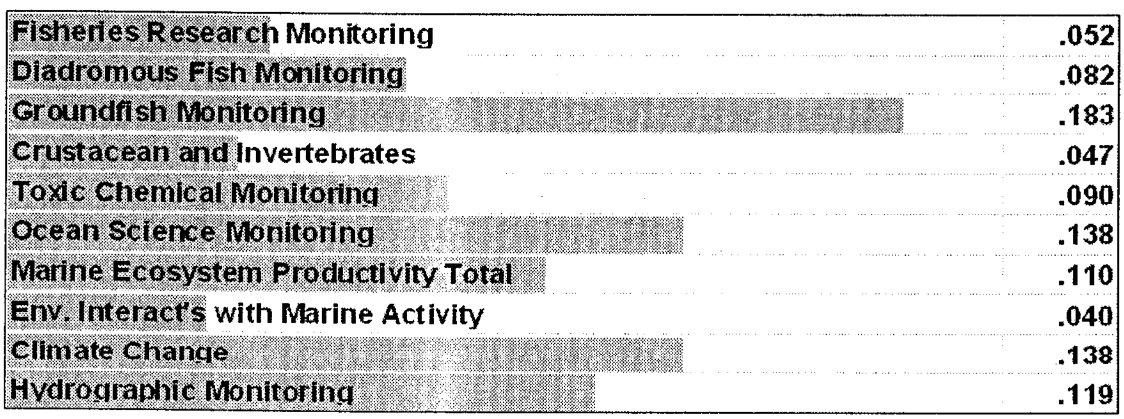
Breakout Group #3 (Figure 5-4). This group favors Ocean Science Monitoring with the highest evaluated priority. Like evaluations for Crustacean and Invertebrates, Diadromous Fish Monitoring and Groundfish Monitoring result in tied second place rankings. The ranking for Hydrographic Monitoring compares closely and other programs all fall somewhat behind. Again, Fisheries Research Monitoring has the lowest priority of all programs.

Figure 5-4: Maritimes monitoring program priorities ranking for group #3



Breakout Group #4 (Figure 5-5). This group clearly favors Groundfish Monitoring in its evaluation for the highly weighted SFA and HaPAE tables. The result is that Groundfish Monitoring achieves the highest priorities in the rankings. Ocean Science Monitoring and Climate Change have identical evaluations resulting in the same second ranked priorities. Hydrographic Monitoring, Marine Ecosystem Productivity Total, Toxic Chemical Monitoring and Diadromous Fish Monitoring rank successively below. Other programs fall still further behind and different with other Groups, Env. Interact's with Marine Activity has the lowest ranked priority ranking.

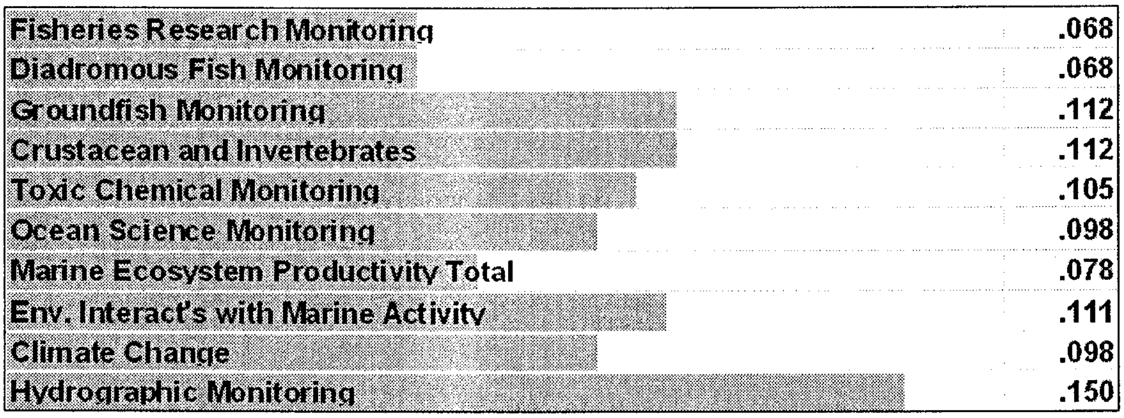
Figure 5-5: Maritimes monitoring program priorities ranking for group #4



Breakout Group #5 (Figure 5-6). This group evaluates Hydrographic Monitoring at the highest possible level resulting in a higher priority than other programs. Fisheries Research Monitoring and Diadromous Fish Monitoring rank lowest.

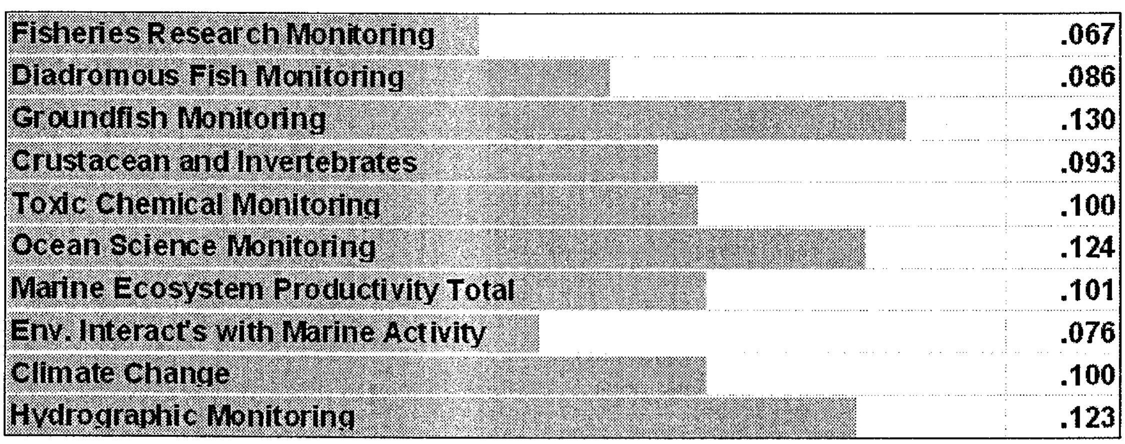
Other programs rank in the middle area with priorities that are comparable.

Figure 5-6: Maritimes monitoring program priorities ranking for group #5



Aggregate Results (Figure 5-7). Overall, the results from the total evaluations of all breakout groups favor Groundfish Monitoring with the highest priority. Ocean Science Monitoring and Hydrographic Monitoring follow close behind. The remaining programs are clumped in priority with Fisheries Research Monitoring and Env. Interacts with Marine Activity with the lowest priority rankings.

Figure 5-7: Aggregate Maritimes monitoring program priorities ranking



5.3.4 Comparison of Modeling Result to Actual Allocation

The resulted aggregated Maritimes monitoring program priorities ranking is then compared with the actual budget allocation from project inventory. The "Original Priorities" shown in Table 5-4 is obtained by the backward calculation illustrated in section 5.2.

The comparison between resulted priorities and original priorities calculated based on the actual budget allocation for aggregated monitoring program shows both the similarities and significant difference. Half of the aggregated monitoring programs' resulted priorities are close to the original one having difference within $\pm 50\%$. On the other hand, there are also half of the programs having significantly different priorities which is greater of less than the original priorities more than 50%.

Table 5-4: Aggregated Maritimes monitoring program result comparison

Monitoring Programs	Original Priorities	Resulted Priorities	Difference
Fisheries Research Monitoring	0.054	0.067	25%
Diadromous Fish Monitoring	0.081	0.086	7%
Groundfish Monitoring	0.146	0.130	-11%
Crustacean and Invertebrates	0.032	0.093	188%
Toxic Chemical Monitoring	0.029	0.100	245%
Ocean Science Monitoring	0.150	0.124	-17%
Marine Ecosystem Productivity Total	0.167	0.101	-40%
Env. Interact's with Marine Activity	0.017	0.076	348%
Climate Change	0.048	0.100	106%
Hydrographic Monitoring	0.276	0.123	-55%

The most prominent characteristic of AHP methodology used in decision making process is its effort to quantify the subjective judgments of decision maker under certain rules during the process. These quantified measures then become objective judgments which can be used to further calculation and therefore make the resulted priorities more mathematically rigorous. But still, the original judgments come from human beings, and different participant would have different point of view when dealing with the same decision problem. Sometime even the same participant would have different ideas when looking at the same problem at a different time. Hence, there is no absolutely optimum result for priorities setting exercise, and the significant difference in the comparison is reasonable since the situation change of the decision making practice. On the other hand, the accuracy in the resulted priorities ranking demonstrates that the participant's knowledge and expertise have been effectively applied to evaluate the expected performance of programs.

5.4 Analysis and Discussion

During the evaluation process by the Breakout Groups, it was noted that there was considerable debate in interpreting the elements of the data worksheet related to the decision hierarchy that was assumed for the Monitoring Programs evaluation. There was also different understanding and confusion related to the definition of risk in the risk-based prioritization practices. The clarification on these concerns is considered an important step for further system development.

5.4.1 Discussion on Problem Formulation

In the discussion of the model structure (Figure 5-1), the three "objectives" were generally accepted as reflections of the national/HQ declaration of the DFO Science primary objectives. These objectives were variously discussed. However, difficulties arose in the interpretation of the "criteria" noted above, namely (1) scientific excellence, (2) long-term efficiency, and (3) decision making applied to the objectives as indicators of delivery.

The origin of these criteria has been discussed in earlier chapter (3.1). One of the difficulties in interpreting the "risk areas" relates to the fact that they are all "absolute" characteristics of science objectives and of program/project delivery. In other words, there is no doubt that the absolute quality of "scientific excellence" should be fully present in all science work. However, for the priority setting exercise, we need to interpret these characteristics not as absolutes, but in terms of their relative importance in achieving the objectives and/or in the successful completion of programs/projects. For example, while scientific excellence is presumed throughout, the need for sound and accurate science information may or may not be as important in one objective and/or program/project as compared to another. Keeping this notion of relative importance in mind will make it easier to evaluate different programs/projects against any decision hierarchy structure.

The inclusion of the criteria of the three strategic objectives stems from the above definition of the "risk areas" (not including the 3 strategic outcomes). As noted above, the "risk areas" are defined as being "common to all Science programs". As such, they are considered to be defining attributes of Science programs/projects. Moreover, it is anticipated that the importance of each common attribute varies relatively as a contributing factor to a specific strategic outcome/objective, i.e., the attribute of scientific excellence for any given project is relatively more or less important for programs/projects that are designed primarily to deliver the SFA objective. Conversely, each program/project can be described by its performance among its ability to deliver on these same attributes.

Thus, the idea is that each strategic objective can be broken down into its component attributes and their relative importance toward delivering the declared objective. At the same time, the performance of projects/programs can be each measured in terms of their attributes' content and coherence with the different objectives. Thus, the criteria provide a link between the objectives and the programs/projects that deliver on these objectives.

There is no real requirement for the criteria to be the same for all objectives in the hierarchy as stated above. The reference in the HQ documents to these attributes being "common to all Science programs" (in the absence of stated attributions to different objectives) provides rationale for the assignment of these same criteria across the different objectives in the above problem formulation. Alternative complete taxonomies for criteria may be defined (although no others were evident from existing DFO literature on Science prioritization).

The problem formulation presented here is not to be considered as being unique. Alternative models may exist by applying different and complete taxonomies of objectives, and associated criteria and sub-criteria that depend on the measures of importance being applied. For example, the criteria used here could be replaced with the suite of science functions. In this case, it becomes necessary to (1) analyze the relative importance of the functions in contributing to the stated objectives (independent of the program/projects), and (2) analyze the functional content of the programs/projects to be prioritized. Together in the alternative problem hierarchy, this information leads to a prioritization of the programs/projects through the objectives and their functions.

5.4.2 Discussion on Risk

Risk management has become an increasingly important practice and one of the significant themes initiated for the changes toward policy and program renewal in DAAP final report (DFO 2003c). "This theme focused on the streamlining of operations and regulations through: Integrating risk management into program management and service delivery strategies; Strengthening management practices, to support the identification of organizational risks and their sound management." Integrating risk into the management framework commit an approach that is more structured, transparent and consistent to program priorities setting based on the significance and/or likelihood of adverse outcomes.

The very first step for risk-based priority setting is to properly define the content of risk analysis associated with it, which is considered lacking in the current management framework. The discussion about the definition of risk analysis motivated by proposed risk analysis methods concentrated on the consequence and cost impact on not meeting the program objective. The consequence refers to not only the ecological influence on the nature, but also socio-economic impacts including partnership, public good, cost-benefit issue, etc. Issues related to this concern were brought forward during the risk-based discussion.

First, risk has been explained in terms of implementation risk related to natural uncertainty and program risk, probability of success and progress. Risk analysis is then interpreted as an approach to avoid the risk by estimating the probability that undesirable outcome of a given program will occur. The estimated probability distribution is then used as basis for the priorities setting. Projects with low probability of success are treated as high-risk projects that may get lower priorities.

Furthermore, when resource is not exchangeable but unexpected event happen, how does cost effect upon decision and how to measure the change and loss were also listed into the issues needed to be refined. These concerns characterize DFO risk-based science prioritization problem with an important feature, information gathering about the uncertainty involved in the program implementation. Cost benefit has always been used as crucial parameters in performance evaluation process and it is necessary to gather these relative feedback and corresponding uncertainty in a dynamic environment. The information would then be used with advantage for the future decision making.

5.5 Workshop Feedback

The DFO risk-based science prioritizations presented in the workshop obtain the most valuable positive feedback from the participants. The proposed model has been considered as an excellent tool for resource allocation planning other than current project inventory approach and also a significant step for planning. As the Region's key decision makers, the invited scientists were encouraged by the proposed methodology and application system.

The application system designed and demonstrated for sample prioritization problem have triggered discussions on various topic interesting individual scientists by its robust methods and understandable user interface. The discussions pointed out certain deficiencies that require future considerations. More objectively, consistent measures and clearer definition of elements within the hierarchy were believed lacking in the current application system. In particular, the decision makers participating in the monitoring program prioritization practice were understandably confused on the term used for criteria.

The following steps toward risk-based science prioritization were then decided in response to the requirement of system improvement and clearer definition on the criteria of hierarchy and associated risk analysis. Other than further refinement on the framework of application system, better defining criteria and programs to discriminate easier, the focus has been set to develop an appropriate approach for risk analysis, which can more effectively incorporate risk measures performance to support future decision making.

5.6 Case Study Extension

This section applies the risk analysis methods, as discussed in Chapter 3 and illustrated in Chapter 4 to the sample model of Maritimes Region monitoring program prioritization practice as the follow-up work to the BIO workshop. The section is considered the extension to complete the experiment present in BIO workshop. The purpose of this extension is to illustrate the expected utility approach proposed in this thesis in response to the discussion about risk analysis during the workshop.

5.6.1 Risk Analysis

Based on the methodology of risk analysis discussed in previous chapters, the process of estimating the probability that a specific program will cause some unexpected outcome exceeding the predefined thresholds is only the first component of risk analysis. Utility is chosen as a more comprehensive measure for risk analysis in this thesis, which represents the total "gain" or "satisfaction" of a program's performance with respect to the given budget and taking all factors into account. The expected utility theory is a broadly accepted approach in risk analysis in decision making and the procedure illustrated in the previous chapters is the simplified case designed especially for DFO risk-based science prioritization.

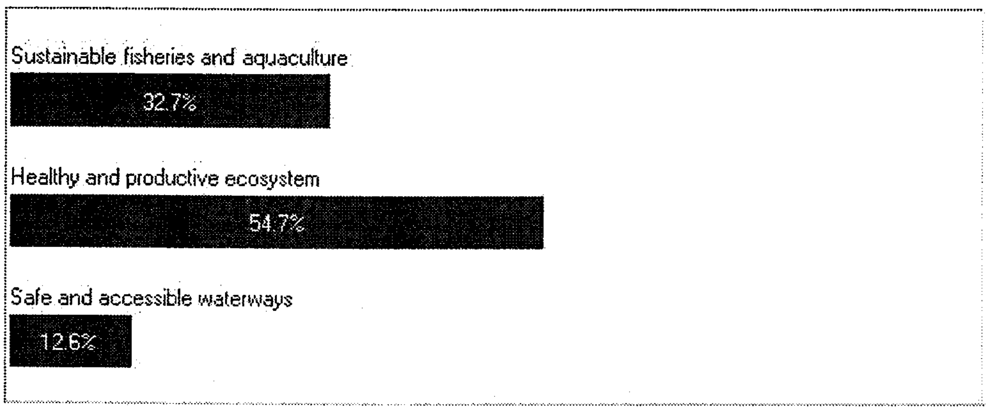
The risk analysis process for risk-based prioritization of Maritimes monitoring programs should consider all possible outcomes so that risk is properly taken into account. In this extension analysis, risk analysis proceeds under a unified and consistent approach integrating the concepts of risk, uncertainty and utility. The prioritization process is referred to the first part of the analysis, risk assessment. The resulting priorities ranking of monitoring programs were then used in a further stage process, risk management, to provide further information supporting the decision making.

5.6.2 Risk Management

As for the situation of the aggregate Maritimes monitoring program priorities setting problem, the prioritization and resulting allocation for a current budget is considered the first step of risk analysis. The result of this process and budget proportion to each program (Figure 5-7) is then saved for the last module of the application system, Risk Analysis. As demonstrated in Chapter 4, risk analysis in this proposed system includes top-down objectives dynamic analysis, bottom-up alternatives' performance dynamic analysis, and finally the utility analysis.

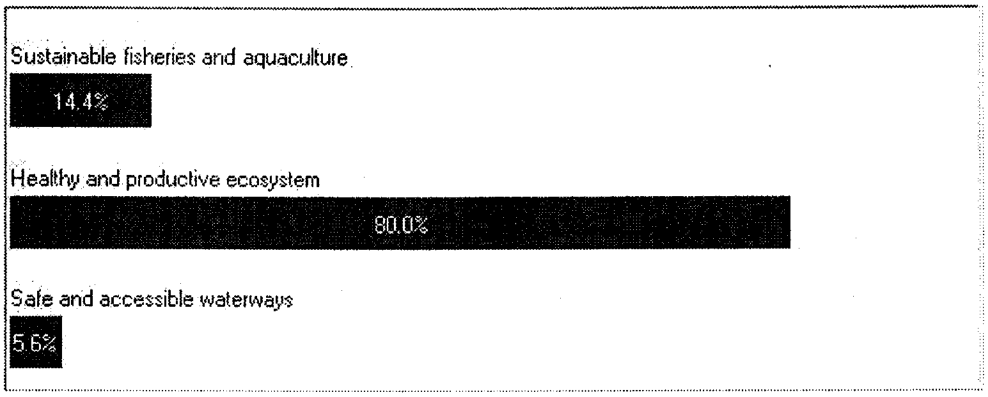
To illustrate the top-down objectives dynamic for the case of Maritimes Region monitoring programs, Figure 5-8 shows the original priorities ranking for the three top objectives of the Maritimes Region monitoring program prioritization model.

Figure 5-8: Original priorities of top objectives



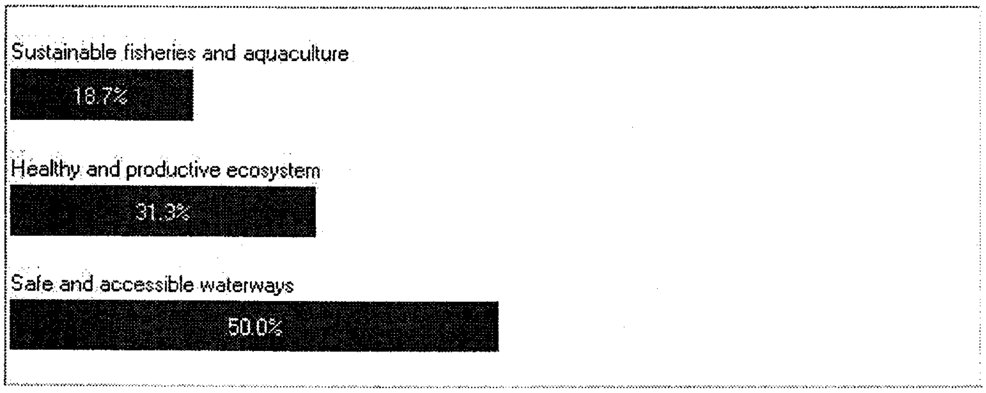
In this first change analysis, the the weight of "Healthy and productive ecosystem" (HaPAE) is changed from 0.547 to 0.8 as shown in Figure 5-9, the priorities of other the two objectives change correspondingly.

Figure 5-9: Priority change for Healthy and Productive Ecosystem



In the second change analysis, the weight of "Safe and accessible waterway" (SAW) is changed from 0.126 to 0.5, the priorities of the other two objectives change as shown in Figure 5-10.

Figure 5-10: Priority change for Safe and Accessible Waterway



The changes in the alternatives' priorities corresponding to the first change in Figure 5-9 are shown in the following Figure 5-11. It is noted that the Toxic Chemical Monitoring program's priority improves since it had a high data grid evaluation value with respect to the HaPAE criteria under this objective. Similarly the priority of the Hydrographic Monitoring program is reduced from 0.122 to 0.102 when the HaPAE criteria are given more importance, since this leads to an overall reduction in the SAW criterion for which Hydrographic Monitoring program is more heavily weighted.

Figure 5-11: Resulting priorities w.r.t the change in HaPAE

(Figure 5-9 vs. Figure 5-8)

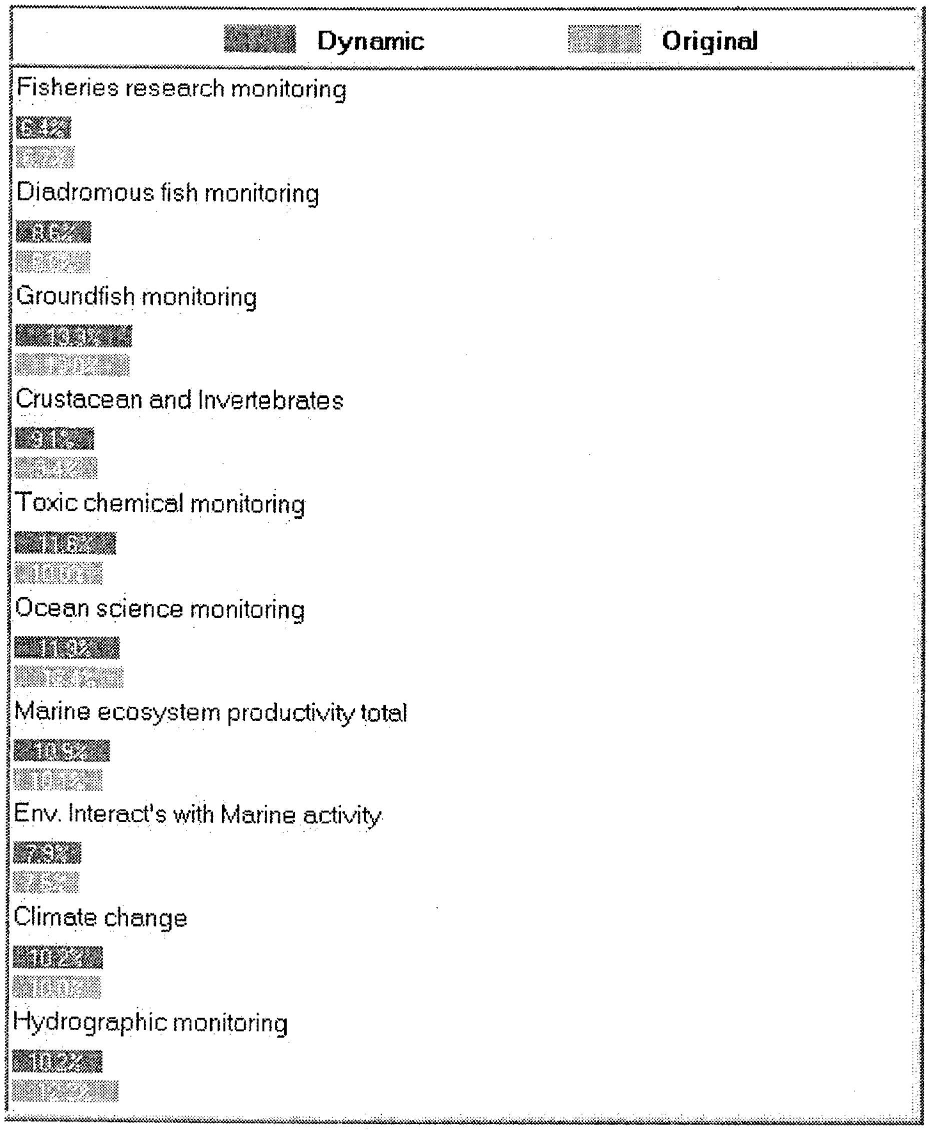
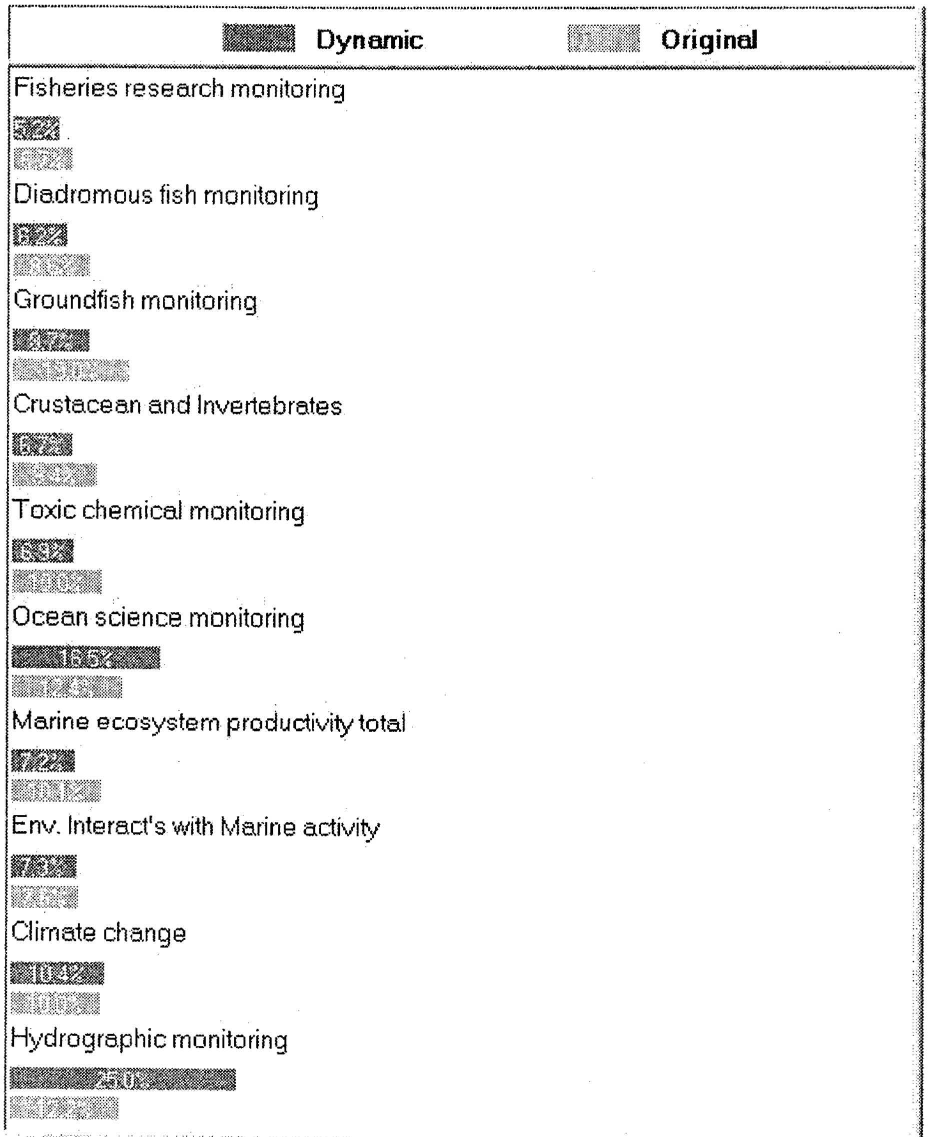


Figure 5-12 shows the resulting priorities' changes due to the increase in the importance of SAW from the original weight of 12.6% (Figure 5-8) to 50% (Figure 5-10). In this case, Ocean Science Monitoring and Hydrographic Monitoring programs' priorities improve significantly since these programs both had high evaluation values with respect to the SAW criterion (Figure 5-12). Similarly the increase in SAW also results in lower priorities notably to Ground fish monitoring, Crustacean and Invertebrates, and the Marine Ecosystem Productivity programs. The Environmental Interactions and the Climate Change programs are relatively unaffected by the SAW increase. This analysis of changes to the top level criteria shows the sensitivities of the program priorities to these changes. Given the potential shifts that may occur in top-down policy, this analysis provides the direct impact of these policy changes.

Figure 5-12: Resulting priorities w.r.t the change in SAW



A second way to observe the allocation system sensitivities is to change the evaluation of the individual programs in terms of increasing or decreasing their datagrid values. As shown in Figure 5-13, if the performance value of the "Groundfish Monitoring Program" with respect to each criterion in the datagrid increases from its original values by 100%, then its priority would go up from

13.0% to 22.7%, and the priorities of other programs will all be adjusted downward accordingly (Figure 5-14). This analysis provides the impact of bottom-up data changes and their impacts on program priorities. The analysis is useful for improving values to satisfy desirable outcomes, or for exploring the uncertainties inherent in the data gathering exercise, especially among multiple participants.

Figure 5-13: Evaluation change of Groundfish Monitoring Program

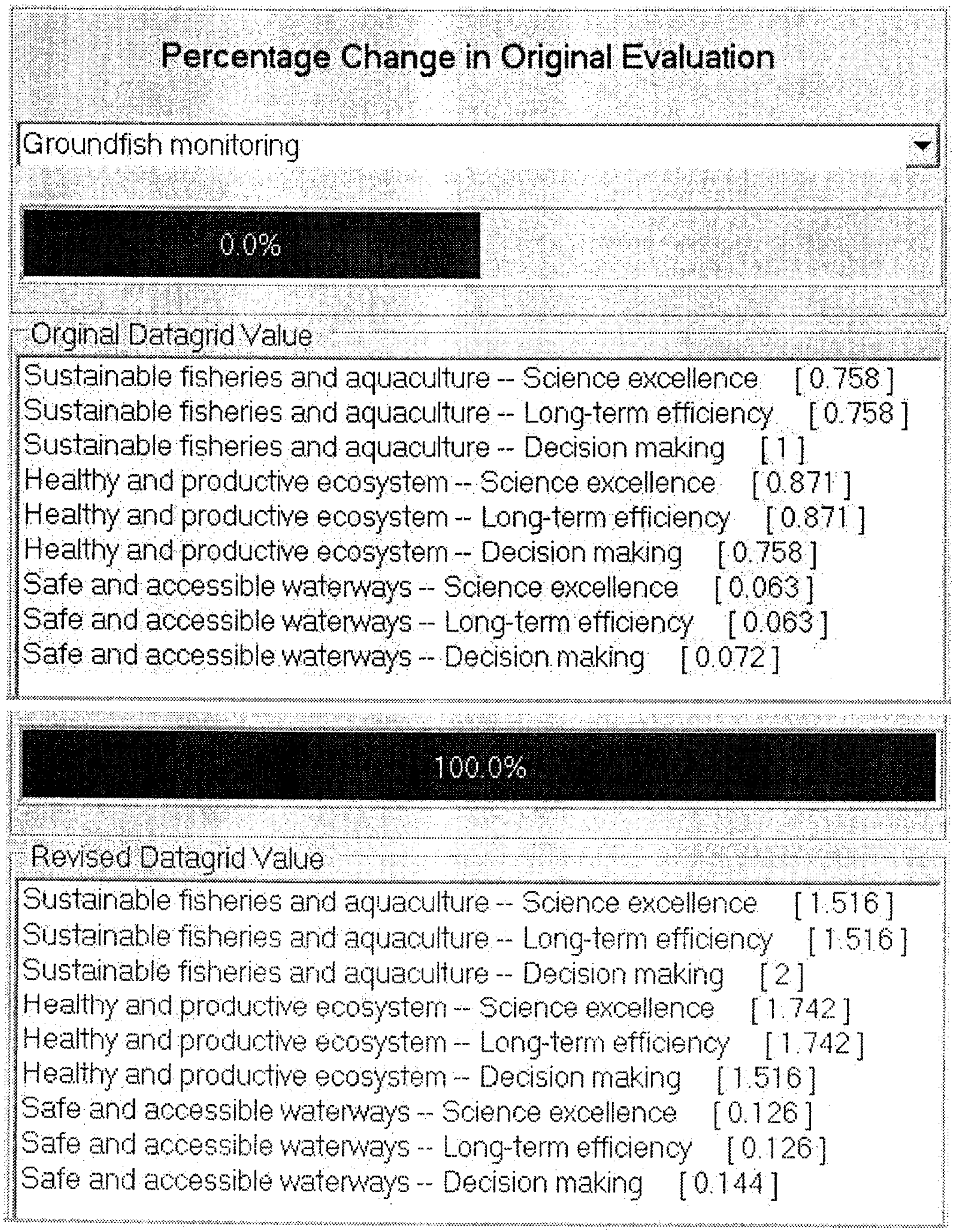
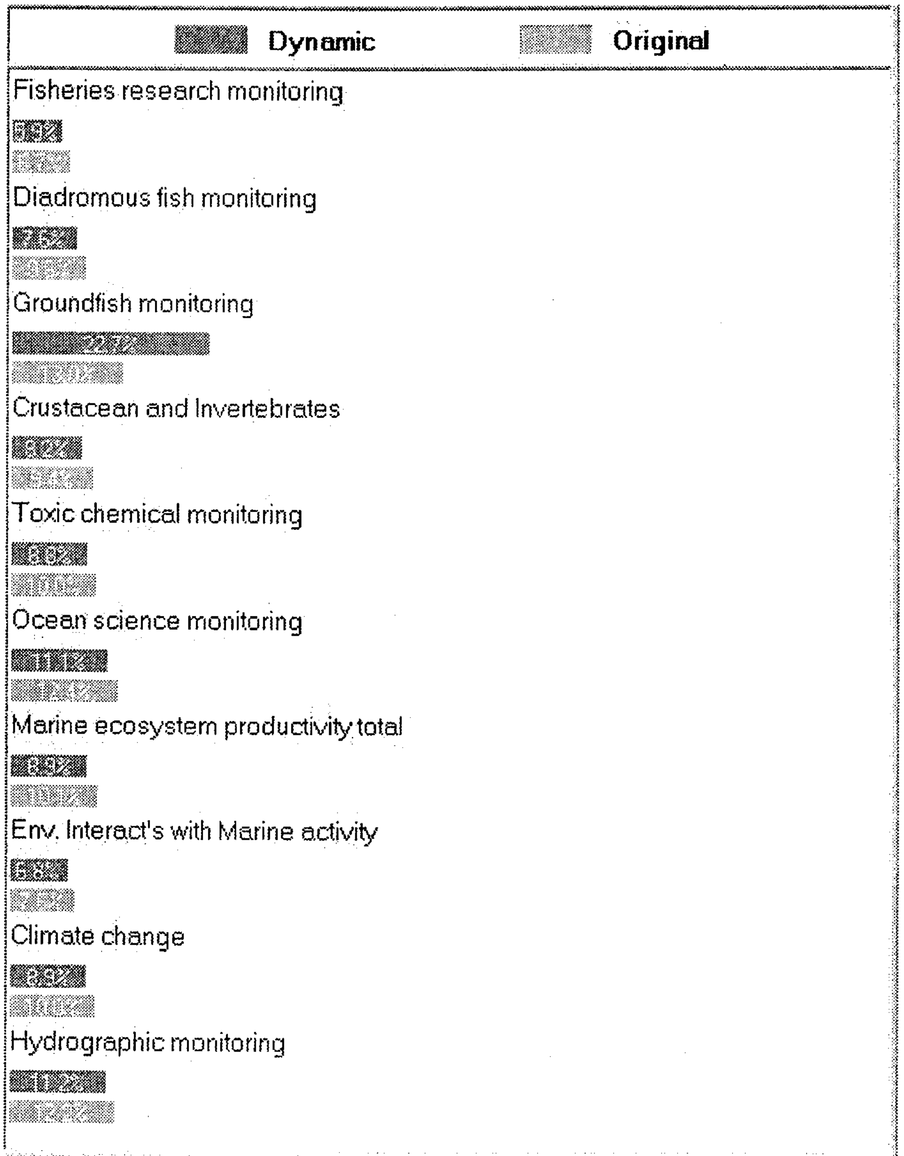


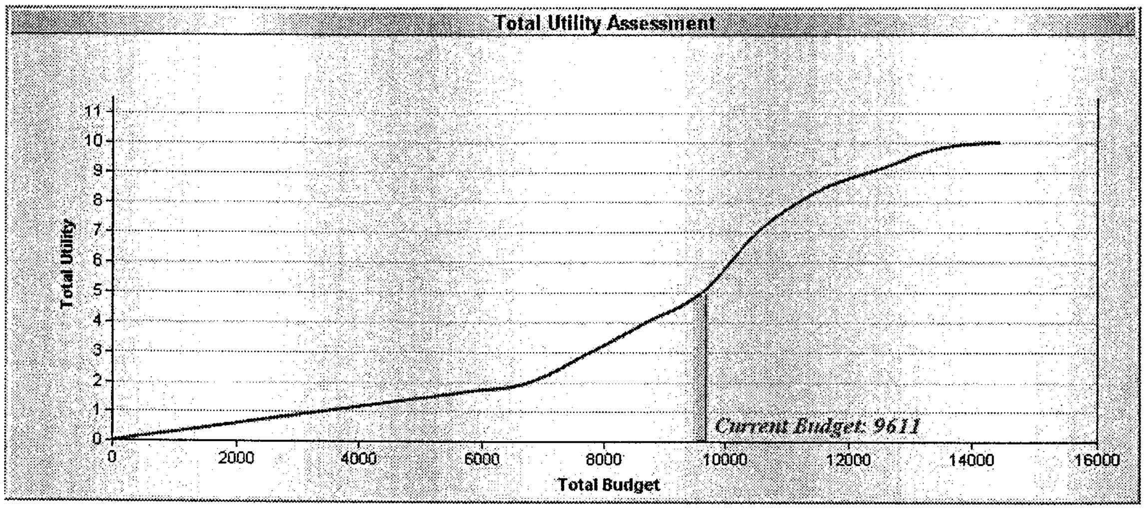
Figure 5-14: Maritimes monitoring program bottom-up dynamic



Finally the last method in the Risk Analysis of the system is the expected utility analysis. Once the low, medium, high and maximum allocation thresholds for each program have been set up (following the method introduced in Chapter 3) the system transforms the resulting priorities of the programs into utility measures. First, the system recognizes where the priority falls between different thresholds and then calculates the corresponding utility for the program using the piecewise linear approximation (Table 3-3). Finally the total utility of all programs is obtained by simply summing the utilities of all programs. It is noted that this simple sum is used to compare overall utilities for different possible thresholds that decision makers may assign, or for different total budgets. The unweighted sum assumes overall utility comes uniformly from all programs.

Suppose the current budget allocations from the prioritization process meet the medium threshold, which means the current budget allocation plan can achieve 50% satisfaction on the programs' performance (Figure 5-15).

Figure 5-15: Maritimes monitoring program current utility – Current budget at programs' medium thresholds



Consequently, changes in the thresholds or changes in the total budget will cause changes in the final utility value. As for the case shown in the figure above, if the Current Budget decreases from \$9611K to \$6000K, the programs' performance as measured by the utility will then fall to an unacceptable level which is down under 40% of the original utility as shown in Figure 5-16.

On the contrary, if instead of reducing the Current Budget, from the graph in Figure 5-15, slightly increasing the Current Budget from \$9611K to \$11000K will result in a significant gain in programs' overall utility performance (Figure 5-17).

Figure 5-16: Resulting utility from a budget reduction for monitoring programs

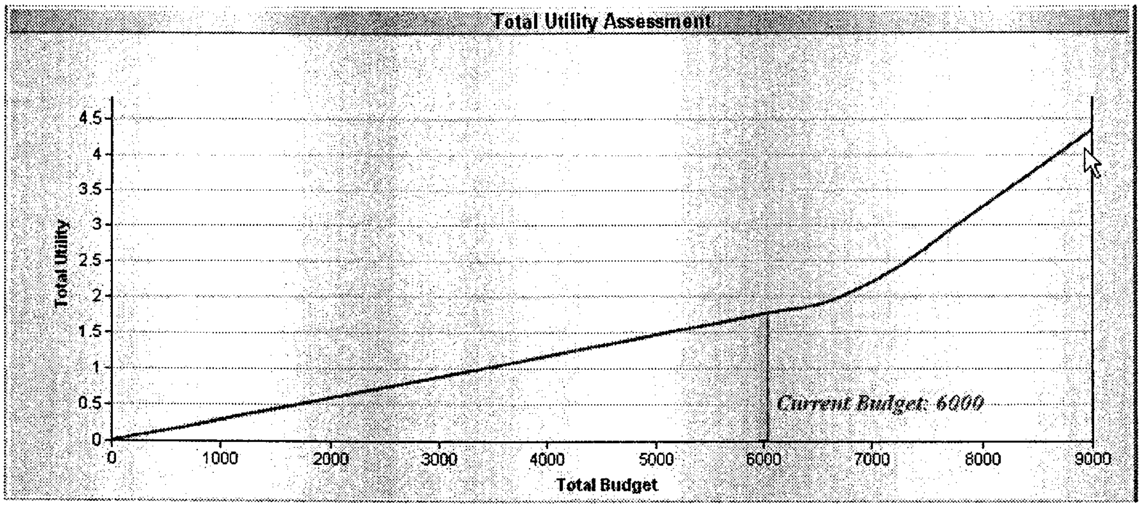
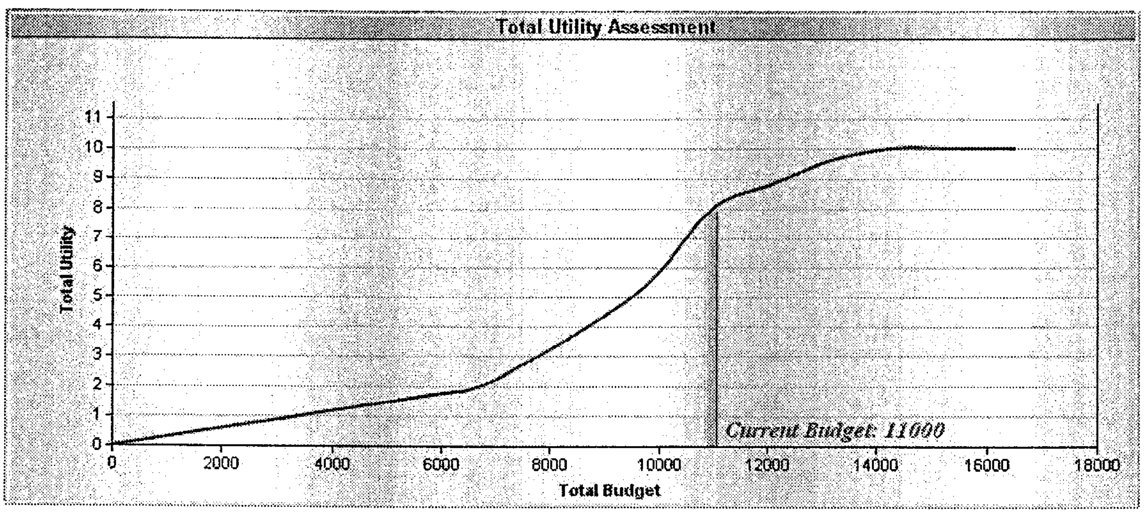


Figure 5-17: Resulting utility from a budget increase for monitoring programs

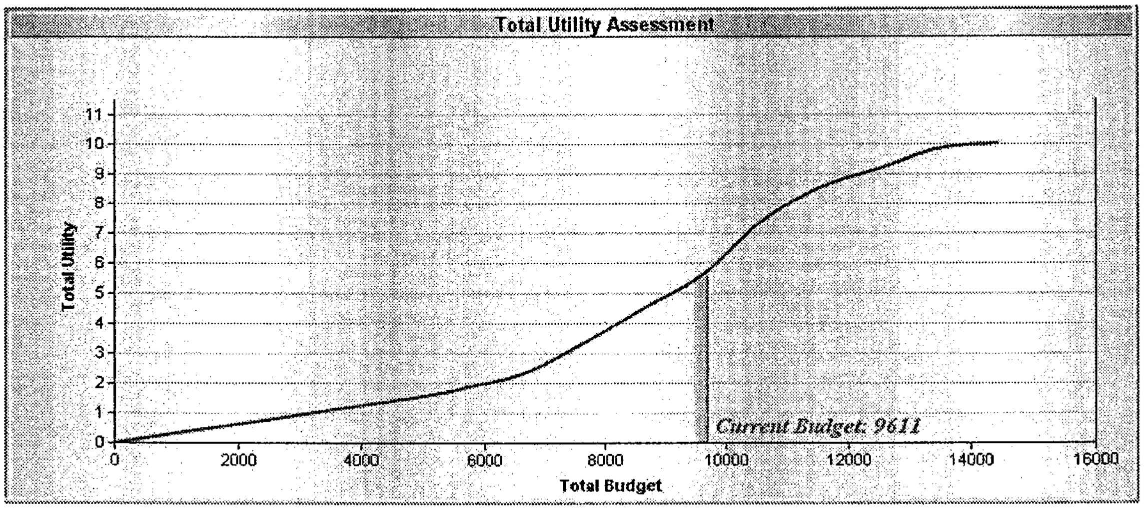


This analysis and the shape of the utility curve provide a basis for arguing for total budgetary adjustments. The slight budgetary increase from $9611K to %11000K represents a 14% increase that earns a positive shift in program performance and delivery as measured by overall utility from 5 to 8 or a 60% improvement in performance, scientific capability, and anticipation of successes for mandate delivery.

Reducing the current thresholds for individual programs, then for the current budget allocation to this program, the same allocation brings in a higher performance. Thus lower thresholds improve the total programs' performance in terms of the utility value. Figure 5-18 shows the result of reducing the thresholds for "Groundfish monitoring" and "Ocean science monitoring" so that the current budget to these two programs fall in their high thresholds, resulting in a higher overall utility value compared to Figure 5-15.

This analysis recognizes the sensitivities related to subjective assignment of programs' threshold values. These values are subject to discussion and the analysis allows for exploring their values and impact.

Figure 5-18: Resulting utility of threshold changes to two monitoring programs



6. Conclusions and Recommendations

In the last Chapter, the outcomes of this thesis research are examined against the objectives of the research stated in Chapter 1. The expected significance of this thesis research is demonstrated by first explaining the model results, and then evaluating and illustrating the results through comparison with the observations of past approaches and existing allocation decisions. Directions for potential future development of the DFO risk-based science prioritization system are also recommended.

6.1 Conclusions

The advantage of using AHP decision methods is the efficiency this model provides to prioritize objectives and projects to simplify and structure complex decision processes, and to provide the ability to track the evaluation and scoring of projects. From the sample model demonstrated in "Modeling and Analysis" and the experiment at the BIO workshop, the comparative analysis with existing budget and allocation practices has proven the efficiency of proposed decision making process. The results of the DFO risk-based science prioritization problem provide alternative and effective solutions for scientific resources allocation designed to achieve the strategic goals of DFO successfully.

- ❖ As a multiple criteria method for science prioritization, AHP provides better understanding of the situation in a decision making problem under risk. The hierarchical framework of the AHP approach contributes most to represent and clarify the problem in a structured mode that makes the prioritization process become more transparent and hence understandable. The pairwise comparison method and matrix calculation contained in the model benefit science prioritization by putting it on a sound structured basis that is more mathematically rigorous.

- ❖ The expected utility theory used in the Risk Analysis module in this thesis has been shown by the illustrations in the thesis that it is a more comprehensive method to measure risk than the commonly held misconception of "risk" as the probability of undesirable event happening.

Associated with this methodology, the risk-based science prioritization is capable of simulating the decision problem under uncertainty to provide sufficient information supporting the future decision making.

- The hierarchy designed for the illustrated DFO prioritization examples incorporates the seven "risk areas" defined by DFO including the three strategic outcomes and four risk associated criteria. It illustrates the crucial contents of DFO risk-based science prioritization and is therefore appropriate for DFO planning and recourse allocation practices.

- The application system developed in this thesis research implement a multiple criteria priority setting process for DFO decision making. Four modules included in the system properly demonstrate the step by step process of the proposed methodology by designing the modules relatively independent of each other through the use of VB.NET programming facilities. The system has included all the needed functions for DFO risk-based science prioritization and the built-in user interface that is easy to use and understand.

6.2 Recommendations

This thesis research is a first step in developing a structured and comprehensive risk-based decision support system for prioritizing science activities. Future research on this problem given the work begun here may be extended in a number of different directions:

- More objective measures can be designed for evaluation and data collection of programs. Since the pairwise comparison and datagrid evaluation are subjected to decision maker's personal experiences and concerns, the resulting priorities ranking is essentially subjective. Therefore, designing sound measures to avoid importing subjectivity during the process should improve the current system. In particular, incorporating the advances in group decision making including the new components of the Expert Choice modules for multiple participants would advance the results of the science prioritization problem.

- A clearer and better defined hierarchy is required from the government mandate. Since the science prioritization model is used for DFO's resource allocation purpose, the hierarchy fit directly into the context of DFO strategic planning. Further research on interpreting the National priorities into Regional operations is evidenced by the current difficulties that their process exhibit in communication between them.

- A more appropriate utility function for risk analysis can be designed from empirical studies for obtaining more accurate utility valuation. The function for calculating the utility of individual alternatives in this thesis is uses piecewise linear approximation for the S-shaped utility curve experimentally. Hence to reduce the uncertainties in utility valuation, a more appropriate utility function could be defined specifically for DFO risk-based science prioritization based on empirical analysis among DFO managers and decision makers.

- More effort can be applied to further development of the application system to suit the group decision making process. The current application system is developed under certain limitations, e.g. the hierarchy can be set no more than four levels and the final priorities ranking is based the judgment of one participant. For the future development, the system can be extended by increasing the size of the hierarchy and creating at the same time the integration of group judgments.

Bibliography

Alston, J. M., Norton, G. W. and Pardey, P. G. 1995. Science under scarcity. Cornell University Press.

Bahcall, J. N., 1991. Prioritizing Scientific Initiatives. Science, 251, 1412-1413. March 22.

Bertsekas, D. P., 1976. Dynamic programming and stochastic control. Academic Press, Inc.

Canada. 2002a. 2002-03 Estimates. A Report on Plans and Priorities. Fisheries and Oceans Canada. http://www.tbs-sct.gc.ca/est-pre/p2_0203e.asp

Canada. 2002b. Fisheries and Oceans Canada Departmental Performance Report for the period ending March 31, 2002. http://www.tbs-sct.gc.ca/rma/dpr/02-03/FO-PO/FO-PO03D_e.asp

Canada. 2003. 2003-04 Estimates. A Report on Plans and Priorities. Fisheries and Oceans Canada. http://www.tbs-sct.gc.ca/est-pre/p2_0304e.asp

CCSP. 2002. Strategic plan for the Climate Change Science Program. Chapter 12. Grand Challenges in Modeling, Observations, and Information Systems. November 19. http://www.climatescience.gov.

Coraza, J., 2003. Allocating resources by setting priorities. The Silverlake Project. Expert Choice, Inc. IBM Case Study from Expert Choice testimonials. http://www.expertchoice.com/Silverlake.htm

DFO. 2000. Setting the course for the new millennium. Science Strategic Plan. Fisheries and Oceans Canada. Ottawa

DFO. 2001. Fisheries and Oceans, Canada national stock assessment review final report. November. Ottawa. Internal document.

DFO. 2002. Science Projects Inventory. Maritimes Region. Science Branch. Fiscal Year 2002-03. August 31, 2002. Dartmouth. Internal regional document.

DFO. 2003a. Ecosystem-based management. A fisheries perspective. February, 2003. Maritimes region. Dartmouth. Internal regional document.

DFO. 2003b. Risk-based approach to priority setting. February, 2003. Maritimes region science retreat. Oceanstone. Dartmouth. Internal regional document.

DFO. 2003c. "Financial stability, program renewal and modern management" The Departmental Adjustment and Alignment Project (DAAP). Final Report. Ottawa. Internal regional document.

DFO. 2003d. Projects Maritimes - by Functions & Tables (Strategic Outcomes) (July 22). July 2003. Dartmouth. Internal regional document.

DFO. 2003e. Model Overview – Risk based Priority Setting Model. October 2003. Ottawa. Internal regional document.

DFO. 2003f. Healthy & productive aquatic ecosystems table. Interim report – Draft. December 2003. Ottawa. Internal regional document.

DFO. 2003g. Fisheries and Oceans Canada Performance Report For the period ending March 31, 2003. http://www.dfo-mpo.gc.ca/publication_e.htm.

Dhaliwal, H. 1999. Speaking notes at La Conférence sur les pêches semi-hauturières du sud du golfe Saint-Laurent, Moncton, N.B. December 6.
http://www.dfo-mpo.gc.ca/media/speech/1999/19991206_e.htm

Expert Choice. 2000a. Resource Allocation - Northeast Fisheries Center. Sample problem dataset. http://www.expertchoice.com/software/group.htm

Expert Choice. 2000b. IBM Case Study from Expert Choice testimonials. Sample problem dataset.
http://www.expertchoice.com/customers/ibm-silverlake.htm

Expert Choice. 2002. VA Case Study from Expert Choice testimonals. Sample problem dataset.
http://www.va.gov/budget/capital/CIM_Guide_FY2002.pdf

Expert Choice. 2003. Expert choice for project prioritization and portfolio management. http://www.expertchoice.com/software/group.htm

Forman, E. H. and Selly, M. A., 2001. Decision by objectives. World Scientific Publishing Co. Pte. Ltd.

ISNAR. 2001. Information and Discussion Forum on Priority Setting in Agricultural Research. http://www.isnar.cgiar.org/fora/Priority/index.htm

Klir, G. J., 1994. Soft computer-aided system theory and technology. CASR '94, 4th International Workshop, Ottawa, Ontario, Canada, May 16-20, 1994, Selected Papers.

Lane, D. E. and Stephenson, R. L. 1998. A framework for risk analysis in fisheries decision-making. ICES Journal of Marine Science, 55: 1-13.

Lane, D. E., 2003. DFO science prioritization. Multicriteria Procedures for Science Program Prioritization at Fisheries and Oceans, Canada. School of Management Working paper. April. 12p.

NASA. 2002. Research Research Maximization And Prioritization (ReMAP) Preface and Executive Summary. Report by the NASA Biological and Physical Research Research Maximization And Prioritization (ReMAP) Task Force to the NASA Advisory Council August 2002. September 10. http://www.spaceref.com/news/viewsr.html?pid=6509

Nicholson, W. K.,1990. Elementary linear algebra with applications. 2nd ed. PWS-KENT Publishing Company.

Pratt, J. W., 1964. Risk aversion in the small and in the large, Econometrica 32 (1964), 122-136.

Saaty, T. L., 1988. The analytic hierarchy process, Eta Services Ltd. (See also http://www.expertchoice.com/hierarchon/references/preamble.htm)

Taylor, B. W., 2001. Introduction to management science. 7th ed. Pearson Education, Inc.

Thibault, R. G., 2002a. Speaking notes for the Minister of Fisheries and Oceans to the Annual Meeting of the Association québécoise de l'industrie de la pêche (AQIP), Saint-Foy Québec. January 24. http://www.dfo-mpo.gc.ca/media/speech/2002/20020124_e.htm

Thibault, R. G., 2002b. Speaking notes at the Standing Committee on Fisheries and Oceans RE: MAIN ESTIMATES. May 23.

Thibault, R. G., 2003. Speaking notes for the Minister of Fisheries and Oceans to the 29th Annual Meeting of the PEI Fishermen's Association, Summerside, P.E.I. January 10. http://www.dfo-mpo.gc.ca/media/speech/2003/20030110_e.htm

Wasyluk, O. J., and Saaty D. 2003. Developing a Portfolio Approach to Capital Investment: A Case Study in Re-Engineering Resource Allocation at the U.S. Department of Veteran's Affairs. Expert Choice, Inc., 2001, posted at DSSResources.COM December 14, 2001

Wymore, A. W., 1993. Synergy: The design of a systems engineering system. CASR '94, 4th International Workshop, Ottawa, Ontario, Canada, May 16-20, 1994, Selected Papers.

Appendix I.

DFO Sample Comparison Forms

To calculate the priorities of each project within the model hierarchy, it is necessary to collect the pairwise comparison result from every participant. The form showing in the next page is carried out as an intermediate report for data collection during AHP model construction for the DFO case.

The contents within this form are filled in based on the model hierarchy established in the first step of proposed science prioritization process. The first part of the form is for the comparison results among the alternative objectives and criteria following the explanation of fundamental scales. The second part of the form is to collect the evaluation data for alternative projects.

Based on this data collection template, six sample data collected from DFO science for the example presented in Chapter 4 are also listed in the following pages.

Science Prioritization Methods _ DFO AHP Example

Participant's ID: ____________________

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	*Definition*	*Explanation*
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua			
Healthy Ecosystems			

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence			
Long-term Efficiency			

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence			
Long-term Efficiency			

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence			
Long-term Efficiency			

Part II: Alternative Absolute Evaluation – Scores Table – enter values in the table below

Alternative Intensity score: Description	*1* Great		*2* Significant		*3* Moderate		*4* Little	*5* Very Little	
Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Alternatives:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**

Science Prioritization Methods _ DFO AHP Example

Participant's ID: Manager1

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	*Definition*	*Explanation*
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua		1/2	2
Healthy Ecosystems			3

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	1/3
Long-term Efficiency			1

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		3	1
Long-term Efficiency			1/3

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	1/5
Long-term Efficiency			1/5

Part II: Program Absolute Evaluation – Scores Table – enter values in the table below

Program Intensity Score: Description	***1*** Great	***2*** Significant	***3*** Moderate	***4*** Little	***5*** Very Little

Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Programs:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**
OBFM/IFMPs	5	3	1	4	2	1	5	4	3
MPAs/MEQ	3	3	1	2	2	1	5	3	3
Species&Habitat Protection	2	1	1	1	2	1	1	2	2
Aquaculture	2	2	1	2	2	1	3	3	2
Hydrography	4	1	1	5	3	3	2	1	1

Science Prioritization Methods _ DFO AHP Example

Participant's ID: Manager2

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	***Definition***	***Explanation***
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua		1	5
Healthy Ecosystems			5

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	1/3
Long-term Efficiency			1/3

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1/3	1/3
Long-term Efficiency			1/3

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		5	1/3
Long-term Efficiency			1/3

Part II: Program Absolute Evaluation – Scores Table – enter values in the table below

Program Intensity Score: Description	***1*** Great	***2*** Significant	***3*** Moderate	***4*** Little	***5*** Very Little

Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Programs:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**
OBFM/IFMPs	4	1	1	2	2	2	4	4	3
MPAs/MEQ	3	4	4	3	3	2	4	3	4
Species&Habitat Protection	1	1	4	2	2	2	3	3	4
Aquaculture	2	4	4	2	2	2	4	3	2
Hydrography	4	3	3	3	3	3	2	1	2

Science Prioritization Methods _ DFO AHP Example

Participant's ID: Manager3

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	***Definition***	***Explanation***
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua		1/2	1/3
Healthy Ecosystems			2

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		2	1/2
Long-term Efficiency			1/2

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		2	1
Long-term Efficiency			1/2

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		2	1/2
Long-term Efficiency			1/2

Part II: Program Absolute Evaluation – Scores Table – enter values in the table below

Program Intensity Score: Description	***1*** Great	***2*** Significant	***3*** Moderate	***4*** Little	***5*** Very Little

Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Programs:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**
OBFM/IFMPs	2	1	1	1	3	1	3	2	1
MPAs/MEQ	1	2	1	1	3	1	5	2	5
Species&Habitat Protection	1	2	1	1	2	2	5	3	5
Aquaculture	1	2	1	1	2	2	5	2	1
Hydrography	1	2	1	1	2	3	1	1	2

Science Prioritization Methods _ DFO AHP Example

Participant's ID: Manager4

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	***Definition***	***Explanation***
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua		1/5	1
Healthy Ecosystems			5

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1/3	1/3
Long-term Efficiency			1

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1/3	1/3
Long-term Efficiency			3

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1/3	1/3
Long-term Efficiency			1

Part II: Program Absolute Evaluation – Scores Table – enter values in the table below

Program Intensity Score: Description	***1*** Great	***2*** Significant	***3*** Moderate	***4*** Little	***5*** Very Little

Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Programs:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**
OBFM/IFMPs	2	1	1	2	1	1	4	3	3
MPAs/MEQ	2	1	1	2	1	1	2	2	2
Species&Habitat Protection	2	2	1	2	1	1	2	3	3
Aquaculture	2	1	1	2	2	2	2	2	1
Hydrography	3	3	3	3	2	2	3	1	1

Science Prioritization Methods _ DFO AHP Example

Participant's ID: Manager5

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	*Definition*	*Explanation*
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua		1	3
Healthy Ecosystems			3

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	3
Long-term Efficiency			1

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	3
Long-term Efficiency			1

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	1
Long-term Efficiency			1

Part II: Program Absolute Evaluation – Scores Table – enter values in the table below

Program Intensity Score: Description	***1*** Great		***2*** Significant		***3*** Moderate		***4*** Little		***5*** Very Little
Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Programs:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**
OBFM/IFMPs	3	3	3	3	3	3	5	5	5
MPAs/MEQ	4	5	4	4	4	4	5	5	5
Species&Habitat Protection	2	2	2	2	3	3	4	4	4
Aquaculture	2	2	2	2	3	3	4	4	4
Hydrography	2	2	2	4	3	3	2	2	2

Science Prioritization Methods _ DFO AHP Example

Participant's ID: Manager6

Part I: Hierarchy Pairwise Comparison – Scores Tables – enter values in the tables below

Comparative Importance	*Definition*	*Explanation*
1	Equally important	Two decision elements (e.g., criteria) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant
2, 4, 6, 8	Intermediate judgment value	Judgment values between equally, moderately, strongly, very strongly, and extremely
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

I.Compare Objectives	Sustainable Fish&Aqua	Healthy Ecosystems	Safe Waterways
Sustainable Fish&Aqua		1/5	5
Healthy Ecosystems			7

IIa. Compare Criteria: Sustainable Fish&Aqua	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		7	5
Long-term Efficiency			1/3

IIb. Compare Criteria: Healthy Ecosystems	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1	5
Long-term Efficiency			7

IIc. Compare Criteria: Safe Waterways	Scientific Excellence	Long-term Efficiency	Decision-making
Scientific Excellence		1/3	1
Long-term Efficiency			1/3

Part II: Program Absolute Evaluation – Scores Table – enter values in the table below

Program Intensity Score: Description	***1*** Great		***2*** Significant		***3*** Moderate		***4*** Little		***5*** Very Little
Objectives:	**Sustainable Fish&Aqua**			**Healthy Ecosystems**			**Safe Waterways**		
Programs:\Criteria:	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**	**SciExc**	**LT eff**	**DM**
OBFM/IFMPs	2	2	1	1	2	1	5	5	5
MPAs/MEQ	2	3	5	2	3	4	4	5	4
Species&Habitat Protection	1	2	2	2	2	4	2	4	3
Aquaculture	3	3	2	4	4	4	5	4	4
Hydrography	3	2	4	4	3	4	5	3	4

Appendix II.

2nd, 3rd, and 4th Level Business Lines (for Maritimes Region Science Projects Inventory)

This electronic document is provided by Sherry Niven and Sharon Morgan from DFO science Maritimes Region. The document provides detail description of Business Lines for Maritimes Region Science Projects Inventory in 2^{nd}, 3^{rd}, and 4^{th} levels. The contents of this document categorize the activities of individual projects following 4 levels Business Lines, which are used for reporting science projects.

Science Projects Inventory Categories of activities used for reporting projects	**Inventaire des projets des Sciences** Catégories des activités selon lesquels les projets sont classifiés
11 Fisheries, Environment & Biodiversity Science	**11 Science halieutique, de l'environnement & de la biodiversité**
111 Fisheries Research **Fisheries Research** Provides the scientific basis for management of the fishery resource for its sustained utilization, including: - stock assessments on all major resource species of fish, aquatic invertebrates, marine mammals and marine plants; - research on assessment methodology and the biology, ecology and dynamics of exploited and related species to develop and improve the scientific knowledge base; provision of scientific advice relevant to the status of resources stocks.	**111 Recherche sur les pêches** La **Recherche sur les pêches** fournit la base scientifique pour la gestion de la ressource halieutique en vue de son utilisation soutenue, et notamment : - l'évaluation des stocks pour les principales ressources de poissons, d'invertébrés aquatiques, de mammifères marins et de plantes marines; - la recherche sur les méthodes d'évaluation et sur la biologie, l'écologie et la dynamique des espèces exploitées et des espèces connexes en vue de développer et d'améliorer les connaissances scientifiques; - la fourniture d'avis scientifiques pertinents au sujet de l'état des stocks des ressources.
1110 - Fisheries Research general Includes management, policy development, planning, coordination and other activities not included in the items listed below.	**1110 - Recherche sur les pêches - généralités** Gestion, élaboration des politiques, planification, coordination et autres activités n'entrant pas dans les rubriques ci-dessous.
1111 - Diadromous and Freshwater Fish Resource assessment and biological research on wild stocks of anadromous, catadromous, and freshwater fish. Species include: salmon, trout, alewife, blueback herring, charr, striped bass, eel, smelt, eulachon, sturgeon, and whitefish.	**1111 - Poissons diadromes et dulcicoles** Évaluation des ressources et recherche biologique sur les stocks sauvages de poissons anadromes, catadromes et dulcicoles. Espèces couvertes : saumon, truite, gaspareau, alose d'été, omble, bar rayé, anguille, éperlan, eulakane, esturgeon, corégone.
1112 - Groundfish Resource assessment and biological research on wild stocks of groundfish. The species include: grey cod, cod, haddock, redfishes, rock fishes, silver hake, white hake, pollack, sablefish, grenadiers, halibuts, flatfishes, cusk, sand eels, wolffish, skates, dogfish, conger eel,	**1112 - Poisson de fond** Évaluation des ressources et recherche biologique sur les stocks sauvages de poisson de fond. Espèces couvertes : morue du Pacifique, morue franche, églefin, sébastes, merlu argenté, merluche blanche, goberge, morue charbonnière, grenadiers, flétans, poissons plats, brosme, lançons, loups, raies,

argentine.	aiguillat, congre, argentine.
1113 - Pelagic Fish Resource assessment and biological research on wild stocks of pelagic fish. The species include: herring, mackerel, tunas, swordfish, shark, capelin.	**1113 - Poissons pélagiques** Évaluation des ressources et recherche biologique sur les stocks sauvages de poissons pélagiques. Espèces couvertes : hareng, maquereau, thons, espadon, requin, capelan.
1114 - Invertebrates and other species Resource assessment and biological research on wild stocks of invertebrates and reptiles. The species include: lobster, crabs, shrimps, prawns, oysters, clams, geoduck, mussels, scallops, squids, octopus, whelks, sea cucumber, sea urchins, corals, marine turtles.	**1114 - Invertébrés et autres espèces** Évaluation de la ressource et recherche biologique sur les stocks sauvages d'invertébrés et de reptiles. Espèces couvertes : homard, crabes, crevettes, huîtres, myes, palourdes, mactres, panope, moules, pétoncles, calmars, pieuvre, buccins, holothuries, oursins, coraux, tortues marines.
1115 - Marine Mammals Resource assessment and biological research on wild stocks of marine mammals. The species include: harp seal, hooded seal, grey seal, harbour seal, ringed seal, bearded seal, California sea lion, steller sea lion, northern elephant seal, walrus, narwhal, beluga, bowhead whale, right whale, minke whale, humpback whale, killer whale, sea whale, harbour porpoise, sea otter.	**1115 - Mammifères marins** Évaluation des ressources et recherche biologique sur les stocks sauvages de mammifères marins. Espèces couvertes : phoque du Groenland, phoque à capuchon, phoque gris, phoque commun, phoque annelé, phoque barbu, otarie de Californie, otarie de Steller, éléphant de mer boréal, morse, narval, béluga, baleine boréale, baleine franche, petit rorqual, rorqual à bosse, épaulard, rorqual boréal, marsouin commun, loutre de mer.
1116 - Marine Plants Resource assessment and biological research on wild stocks of marine plants. The species include: Irish moss, laver, other reds, kelp, *Laminaria*.	**1116 - Plantes marines** Évaluation des ressources et recherche biologique sur les stocks sauvages de plantes marines. Espèces couvertes : mousse d'Irlande, porphyre, autres algues rouges, laminaires.
112 Environmental Science **Environmental Science** provides the scientific basis for the conservation and protection of fish, fish habitat, and marine ecosystems and the sustainable utilization of aquatic resources. Includes research and understanding of the capacity of fish habitats to sustain fish production, and of the effects on the ecosystem of human activity impacts, such as the physical disruption of fish habitat, the introduction of contaminants, and the unintentional introduction of aquatic invasive species._	112 Science de l'environnement Les **sciences de l'environnement** servent à établir des fondements scientifiques pour la conservation et la protection du poisson, de son habitat et des écosystèmes marins ainsi que pour l'utilisation durable des ressources aquatiques. Y est compris la recherche et la compréhension de la capacité de production des habitats du poisson et des répercussions sur l'écosystème de l'effet des activités humaines, comme la perturbation physique de l'habitat du poisson, l'introduction de contaminants et l'introduction involontaire d'espèces aquatiques envahissantes.

1120 - Environmental Science, General Management, policy development, planning, coordination, and other activities not included in the categories listed below.	**1120 - Sciences de l'environnement - généralités** Gestion, élaboration de politiques, planification, coordination et autres activités ne touchant pas aux catégories ci-dessous.
1121 - Toxic Chemicals Fate in aquatic ecosystems and biological effects of toxic chemicals on fish, fish habitat, and aquatic ecosystems.	**1121 - Évaluation des produits chimiques toxiques** Destin dans les écosystèmes aquatiques et effets biologiques des produits chimiques toxiques sur le poisson, l'habitat du poisson et les écosystèmes aquatiques.
1122 - Aquatic Invasive Species Fate and ecosystem impacts of unintentionally introduced aquatic invasive species, including studies on the environmental effectiveness of various management/control actions.	**1122 - Espèces envahissantes aquatiques** Destin et répercussions sur l'écosystème des espèces envahissantes aquatiques involontairement introduites, y compris des études sur l'efficacité de diverses mesures de gestion et / contrôle sur l'environnement.
1123 - Fish Habitat Capacity of fish habitats to sustain fish production. Effects of human activity (excluding toxic chemical impacts) on fish, fish habitat, and aquatic ecosystems.	**1123 - Habitat du poisson** Capacité de production des habitats du poisson. Effets des activités humaines (à l'exception des répercussions chimiques) sur le poisson, l'habitat du poisson et les écosystèmes aquatiques.
1124 - Environmental Effects of Aquaculture Effects of finfish and shellfish aquaculture on aquatic ecosystems, including the potential near-end and far-field effects of wastes and the ecosystem impacts of chemicals used by the industry (excluding introduced species). (Cross reference with **1225: Aquaculture-Environment Interactions**).	**1124 - Impact environnemental de l'aquaculture** Effets de l'aquaculture de poissons, de mollusques et de crustacés sur les écosystèmes aquatiques, y compris les répercussions éventuelles sur le milieu immédiat et lointain de déchets et les effets sur l'écosystème des produits chimiques qu'utilise l'industrie (à l'exception d'espèces introduites). (À comparer avec **1225 : Interactions aquaculture-environnement**).
1125 - Environmental Data & Advice Provision of scientific data, information and advice on productive capacity of fish habitats, fish habitat impact assessments and, the state and condition of the aquatic environment. Formal environmental impact assessments and the participation in formal scientific peer-reviews (i.e. regional and national advisory processes).	**1125 - Données et avis sur l'environnement** Prestation d'avis, de renseignements et de données scientifiques sur la capacité de production des habitats du poisson, sur les répercussions touchant l'habitat du poisson et sur l'état du milieu aquatique. Évaluations officielles des répercussions environnementales et participation à des examens scientifiques officiels par des pairs (c.-à-d. des processus consultatifs régionaux et nationaux).
113 Biodiversity Science Coordination of activities required to fulfilling the science component of Canada's commitments under the Convention on Biological Diversity.	**113 Science de la biodiversité** Coordination des activités essentielles pour la réalisation de la composante scientifique des engagements Canadiens dans le cadre de la Convention sur la diversité biologique.

1131 - Biodiversity Science All activities relating to implementing the Canadian Biodiversity Strategy (Aquatic Module), to fostering development of knowledge and information on aquatic biological diversity in Canada. Also includes management, policy development, planning, coordination and other activities not included in the other items.	**1131 - Science de la biodiversité** Toutes les activités se rapportant à la réalisation de la Stratégie Canadienne de la biodiversité (module aquatique), pour stimuler la production de connaissances et d'informations sur la diversité biologique aquatique au Canada, y compris l'élaboration de politiques, la planification, la coordination et autres activités non-incluses dans les autres rubriques.
1132 - Species at Risk All activities relating to DFO' Species at Risk Program, including overall coordination of the Program, preparation of Response Statements and Recovery Strategies for *endangered*, *threatened* and *special concern* species, and to the implementation of Recovery Action Plans and projects pursuant to the Strategies. Particularly includes research related to and assessment of species at risk and liaison with the Committee on the Status of Endangered Wildlife in Canada [COSEWIC]. Also includes items relating to the Habitat Stewardship Program and the Interdepartmental Recovery Program.	**1132 - Espèces en péril** Toutes les activités se rapportant à la réalisation du programme du MPO sur les espèces en péril, y inclus la coordination globale du programme, formulation de prise de position et de stratégies de recouvrement pour les espèces en danger, dans un état alarmant ou inquiétant, et à l'exécution des plans et projets de recouvrement prévus selon les stratégies. Cela comprend en particulier la recherche et l'évaluation liées aux espèces en péril, la liaison avec le Comité sur la situation des espèces en péril au Canada (COSEPAC) ; les activités liées au programme d'intendance de l'habitat et le programme de recouvrement interministériel.
1133 - Ecosystem Science Activities relating to the science needed to support ecosystem-based management, including studies on cumulative human impacts on oceans and on ecosystem impacts of fishing.	**1133 - Science des écosystèmes** Activités liées à la science nécessaires pour appuyer la gestion basée sur les écosystèmes, y inclus les travaux d'étude sur l'impact cumulatif des activités humaines sur les océans et l'impact de la pêche sur les écosystèmes.
12 Oceans & Aquaculture Science	**12 Science océanique & aquicole**
121 Oceans Science **Oceans Science** provides the physical, chemical and biological data, analysis and advice necessary to describe the marine environment in support of sustainable development, including:	**121 Sciences des océans** Les **Sciences des océans** fournissent les données, les analyses et les avis nécessaires, dans les domaines de la physique, de la chimie et de la biologie, pour décrire le milieu marin dans la perspective du développement durable, et notamment :

- environmental monitoring and prediction of ocean variables needed to assess linkages to, and impacts of, resource development and conservation, such as temperature, salinity, productivity, etc., their variability and trends; - research on oceanographic processes important in the understanding of environment/fisheries interactions, such as circulation, upwelling, water mass movement, etc.; - provision of ocean data, information and advice to those industries operating in the marine environment, such as fisheries, frontier oil and gas, marine transportation and coastal development.	- la surveillance environnementale et la prédiction des variables océaniques nécessaires pour évaluer les liens et les impacts de l'exploitation et de la conservation des ressources, notamment la température, la salinité, la productivité, etc., ainsi que leur variabilité et leurs tendances; - la recherche sur les processus océanographiques importants pour comprendre les interactions environnement-pêches, notamment la circulation, les remontées d'eau, le mouvement des masses d'eau, etc.; - la fourniture de données, d'information et d'avis océanologiques aux industries qui opèrent dans le milieu marin, notamment les pêches, la prospection du pétrole et du gaz, le transport maritime et le développement du littoral.
1210 - Oceans Science general Management functions or generic issues related to several 4th level work activities.	**1210 - Sciences des océans, généralités** Fonctions de gestion ou questions génériques touchant plusieurs activités de quatrième niveau.
1211 - Physical & Chemical Oceanography, Living Resources Physical and chemical oceanography in support of living resources, including environmental monitoring, process research, instrument development, modelling, etc.	**1211 - Océanographie physique et chimique et Ressources vivantes** Océanographie physique et chimique dans la perspective des ressources vivantes, notamment surveillance environnementale, recherche sur les processus, développement d'instruments, modélisation, etc.
1212 - Biological Oceanography/Ecology Biological oceanography and ecological research in support of living resources, including environmental monitoring, process research, instrument development, modelling etc.	**1212 - Océanographie biologique, écologie** Océanographie biologique et recherche écologique dans la perspective des ressources vivantes, notamment surveillance environnementale, recherche sur les processus, développement d'instruments, modélisation, etc.
1213 - Environmental Interactions with Marine Activity	**1213 - Interactions de l'environnement avec l'activité marine**
1214 - Ocean Climate Monitoring of the ocean environment to describe its average and extreme conditions, its seasonal, yearly and long term variability, and its trends or changes with time. Included are carbon cycle research and study of large scale forcing functions that effect the coastal	**1214 - Climat océanique** Surveillance du milieu océanique pour décrire ses conditions moyennes et extrêmes, sa variabilité saisonnière, annuelle et à long terme, et ses tendances ou son évolution avec le temps. Couvre la recherche sur le cycle du carbone et l'étude des fonctions de forçage à grande échelle qui ont un effet sur

environment.	l'environnement côtier.
1215 - Ocean Data, Information, & Advisory Services Includes all aspects of ocean data capture, archiving and delivery, such as quality control, data access, etc. Information products from archived data, regular modelling outputs and advisory services.	**1215 - Données/services de conseil en océanologie** Tous les aspects de la collecte de données océanologiques, de leur archivage et de leur diffusion, comme le contrôle de la qualité, l'accès aux données, etc. Couvre aussi les produits d'information tirés des données archivées, les productions régulières de la modélisation et les services de consultation.
1216 - Harmful Marine Algae Laboratory and field studies on phycotoxins and other harmful effects of marine algal blooms and the organisms producing them; environmental factors and bloom forecasting; biological pathways of toxins, their fate and effects on the food web.	**1216 - Algues marines nuisibles** Études de laboratoire et de terrain sur les phycotoxines et les autres effets dangereux des proliférations d'algues marines, et sur les organismes qui les produisent; facteurs environnementaux et prévision des proliférations; cheminement biologique des toxines, leur devenir et leurs effets sur le réseau trophique.
122 Aquaculture Science	**122 Sciences aquacoles**
Research, experimental development and scientific advice and services relevant to: - the intensive production of fish, aquatic invertebrates and marine plants; - restoring and augmenting the production of wild stocks; - curbing the introduction and spread of communicable fish diseases.	Recherche, développement expérimental, avis scientifiques et services concernant : - la production intensive de poissons, d'invertébrés aquatiques et de plantes marines; - le rétablissement et l'accroissement de la production des stocks sauvages; - la lutte contre l'introduction et la diffusion des maladies transmissibles du poisson.
1220 - Aquaculture Science, general Includes policy development, planning, coordination and technology transfer; international, regional and HQ aquaculture coordination, management and advice; other activities not included below.	**1220 - Sciences aquacoles - généralités** Couvre la gestion, l'élaboration des politiques, la planification, la coordination et le transfert technologique; la coordination, la gestion et les avis en matière d'aquaculture au niveau international, au niveau régional et à l'administration centrale; d'autres activités n'entrant pas dans les rubriques ci-dessous.
1221 - Fish Health Prevention, control and eradication of diseases communicable to aquatic organisms; administration, laboratory and field work related to the FISH HEALTH PROTECTION REGULATIONS for finfish and shellfish including issuing Import Permits; protocols, diagnostics and diagnostic methodologies; research on diseases and parasites of aquatic organisms; chemotherapy and drug/pesticide research and evaluations; quarantine systems.	**1221 - Santé du poisson** Prévention, lutte et élimination des maladies transmissibles aux organismes aquatiques; administration, travail en laboratoire et sur le terrain en rapport avec le RÈGLEMENT SUR LA PROTECTION DE LA SANTÉ DU POISSON pour les poissons et les invertébrés, y compris la délivrance des permis d'importation; protocoles, diagnostics et méthodes de diagnostic; recherche sur les maladies et les parasites des organismes aquatiques; recherche et évaluation sur les

	chimiothérapies, les médicaments et les pesticides; systèmes de quarantaine.
1222 - Invertebrate Aquaculture Nutrition, physiology, genetics, endocrinology, feed technology and food standards, culture systems, larval, juvenile and adult rearing, biotechnology, transgenics, performance testing, cryopreservation, management practices.	**1222 - Aquaculture des invertébrés** Nutrition, physiologie, génétique, endocrinologie, technologie des aliments et normes, systèmes de culture, grossissement des larves, des juvéniles et des adultes, biotechnologie, organismes transgéniques, contrôle de performance, cryopréservation, méthodes de gestion.
1223 - Finfish Aquaculture Nutrition, physiology, genetics, biotechnology, transgenics, endocrinology, feed technology and food standards, culture systems, larval, juvenile and adult rearing, performance testing, cryopreservation, management practices.	**1223 - Aquaculture des poissons** Nutrition, physiologie, génétique, biotechnologie, organismes transgéniques, endocrinologie, technologie des aliments et normes, systèmes de culture, grossissement des larves, des juvéniles et des adultes, contrôle de performance, cryopréservation, méthodes de gestion.
1224 - Introductions and Transfers Committees Assessment of potential impacts and interactions of species proposed for introduction or transfer into Canada, between provinces or watersheds within a province. Includes consideration of species exotic or non-indigenous (new) to Canada, a province or watershed, or different races of species, which already occur (indigenous). May also provide advice on transgenics. Excludes consideration of accidentally introduced species.	**1224 - Comités sur les introductions et les transferts** Évaluation des impacts potentiels et des interactions des espèces, dont l'introduction ou le transfert, sont proposés au Canada, entre les provinces ou entre les bassins versants d'une province. Comprend l'examen des espèces exotiques ou non indigènes (nouvelles) au Canada, dans une province ou un bassin ou de différentes races d'espèces déjà présentes (indigènes). Peut aussi couvrir les avis sur les organismes transgéniques. Ne comprend pas l'examen des espèces introduites accidentellement.
1225 - Aquaculture-Environment Interactions Laboratory and field studies to assess the two-way interactions between aquaculture and the environment, e.g. impact of man-made chemicals on aquaculture operations, impact of uneaten feed and faecal deposition on the environment.	**1225 - Interactions aquaculture-environnement** Études de laboratoire et de terrain visant à évaluer les interactions entre l'aquaculture et l'environnement, tel que l'effet des produits chimiques anthropiques sur les opérations aquacoles, l'impact des aliments non consommés et du dépôt de fèces sur l'environnement.
1226 - Enhancement (including hatcheries) Operating finfish hatcheries, restoring and augmenting the production of wild stocks, providing advice and other scientific services for enhancement, streamside incubators etc.	**1226 - Mise en valeur (y compris les écloseries)** Exploitation d'écloseries de poissons, rétablissement et accroissement de la production des stocks sauvages, fourniture d'avis et d'autres services scientifiques pour la mise en valeur, incubateurs au fil de l'eau, etc.

13 Canadian Hydrographic Service	**13 Service Hydrographique du Canada**
131 Hydrography Conducting field surveys and gathering relevant data on tide, water levels and currents, and compiling and publishing accurate charts and navigational publications of Canadian and adjacent and international waters. Also, participation in geophysical mapping for offshore development and for maritime boundary negotiations. The development of technology to increase the accuracy and efficiency of data collection and chart production.	**131 Hydrographie** Réalisation de levés sur le terrain et la collecte de données pertinentes sur les marées, les niveaux de l'eau et les courants, ainsi que la compilation et la publication de cartes exactes et de documents pour la navigation dans les eaux canadiennes et les eaux internationales proches. Participation à la cartographie géophysique en vue du développement des zones extracôtières et des négociations sur les frontières maritimes. Mise au point de la technologie permettant d'accroître l'exactitude et l'efficacité de la collecte des données et de la production des cartes.
1310 – Hydrography, General All aspects of hydrography, including program management, not contained under the headings below.	**1310 - Hydrographie - généralités** Tous les aspects de l'hydrographie, notamment la gestion des programmes, qui n'entrent pas dans les rubriques ci-dessous.
1311 - Hydrographic Surveys All items related to the data-gathering mission.	**1311 - Levés hydrographiques** Tous les aspects de la collecte des données.
1312 - Hydrographic Products All items related to CHS products and services.	**1312 - Produits hydrographiques** Tous les aspects des produits et services du SHC.
1313 - Offshore Mapping All items related to the CHS Ocean Mapping program.	**1313 - Cartographie des zones extracôtières** Tous les aspects du programme de cartographie des océans du SHC.
1314 - Technology Applications & Transfers Partnering and all items related to the development of new applications and their subsequent transfer to Industry.	**1314 - Applications et transferts technologiques** Création de partenariats et tous les aspects du développement de nouvelles applications et de leur transfert ultérieur à l'industrie.
14 Special Programs	**14 Programmes spéciaux**
141 Sea Lamprey	**141 Lamproie de mer**
1411 - Sea Lamprey Control Program	**1411 - Programme de contrôle de la Lamproie de mer**
142 Vessel Support to Others	**142 Soutien de Navires à d'autres**

1421 - Vessel Support to NSERC	**1421 - Soutien de Navires au CNRG**
1422 - Vessel Support to NRCan	**1422 - Soutien de Navires à RNCan**
1423 - Vessel Support to Memorial University	**1423 - Soutien de Navires à Memorial University**
1424 - Vessel Support to Env. Canada	**1424 - Soutien de Navires à Env. Canada**
1425 - Vessel Support to Fisheries Mgmt.	**1425 - Soutien de Navires à Gestion des pêches**
15 Program Management	**15 Gestion des programmes**
151 Program Management & Technical Support Management, administrative and coordination activities, technical support and vessel support provided to all work activities of Science or on behalf of the Science Sector as a whole.	**151 Gestion des programmes et soutien technique** Gestion, administration et coordination, soutien technique et soutien aux navires dans les secteurs d'activité des Sciences ou au nom de l'ensemble du secteur des Sciences.
1510 - Executive & Regional Science Management ADM, Science and office staff, Regional Directors of Science and office staff, Program Planning and Coordination Branch (HQ).	**1510 - Gestion des Sciences, Haute direction et Régions** SMA, Sciences et personnel de son cabinet, directeurs régionaux des Sciences et leur personnel, Direction de la planification des programmes et de la coordination (AC).
1511 - Management & Support, General Management, administrative or coordination activities provided to or on behalf of all Science work activities which do not fall into one of the items listed below.	**1511 - Gestion et soutien technique - généralités** Activités de gestion, d'administration ou de coordination, réalisées au profit ou au nom de toutes les activités du secteur des Sciences, qui n'entrent pas dans les rubriques ci-dessous.
1512 - Technical Support Technical support services provided to more than one 2nd level work activity, including scientific computing, library services and the Technology Transfer Office in PPC (HQ).	**1512 - Soutien technique** Services de soutien technique offerts à plus d'une activité de second niveau, notamment le calcul scientifique, les services de bibliothèque et le bureau du transfert de technologie à PPC (AC).
1513 - Vessel Support Maintenance and support for ship and launch time obtained from the Canadian Coast Guard or charters for science activities.	**1513 - Soutien des navires** Entretien et appuie pour l'accès aux bateaux et vedettes, de la Garde côtière Canadienne ou de bateaux affrétés pour les activités scientifiques.

Appendix III.

Maritimes BIO Monitoring Program Risk-based Prioritization Report

This report summarizes the results of the Breakout Group evaluation and feedback for designated monitoring programs in the Maritimes region. The five (5) Breakout Groups defined at the Workshop were charged with evaluating an "importance" grid for each of the 10 aggregated Monitoring Programs defined below. This importance grid required that each monitoring program be evaluated relative to defined objectives and criteria for the delivery of science programs.

Monitoring Programs Definitions. The suite of Science project inventory programs designated with at least 50% contribution to the "Monitoring" function were aggregated into 10 programs as described below:

Aggregate Monitoring Program	*Description*
1. Fisheries Research Monitoring	Includes International Observer Program, National Sampling Program, Oceanographic Data Handling
2. Diadromous Fish Monitoring	Includes DFO-NSPC Partnering Agreement, IRF IBOF 3-Rivers Monitoring, Mactaquac Fish Collection Facility, Maritimes Non-Salmon, Nova Scotia and Inner Bay of Fundy Salmon, South and Western N.B. Salmon
3. Groundfish Monitoring	Includes Groundfish Trawl Surveys, Sentinel Survey - Scotian Shelf
4. Crustacean and Invertebrates	Includes Crustacean Sampling Databases, Full Bay Scallop Association FY 02-03, Offshore Clam Research Project, Research Coordination/Gulf of Maine Crustacean Fisheries Section, Sea Cucumber 4X/5Z
5. Toxic Chemical Monitoring	Includes Clay-Oil Flocculation Monitoring, Point Lepreau Environmental Monitoring Program
6. Ocean Science Monitoring	Includes Continuous Plankton Recorder (CPR) Sampling Program, Section Overhead, Shelf Climate Variability
7. Marine Ecosystem Productivity Total	Includes three Marine Ecosystem Productivity projects
8. Env. Interact's with Marine Activity	Includes Brooke Ocean Technology, Coastal Habitat Studies
9. Climate Change	Includes Climate Change Impact on Energy Sector, Coastal Ocean Climate
10. Hydrographic Monitoring	Includes Hydrography General, Labrador Survey - PI 2, Multibeam Seafloor Mapping, Terra Remote Sensing Inc.

Monitoring Programs Input Evaluation. Each breakout group was required to complete an evaluation of the 10 Monitoring Programs on an importance grid with respect to the contribution of each program to the science objectives: (1) Sustainable Fisheries and Aquaculture; (2) Healthy and Productive Aquatic Ecosystems; and (3) Safe and Accessible Waterways and the objectives' criteria of Scientific Excellence, Long-term Efficiency, and Sound Decision-making. Discrete measures of 1 to 5 were used, where 1 denoted greatest level of importance and 5 denoted least level of importance of each program in contribution to each combination of objectives and criteria. Each Breakout Group's feedback datasheet are provided below along with a summary graphic report of these Group results. Aggregate results summarizing all Breakout Groups' feedback are also provided as an average evaluation for each cell of the datasheet.

Monitoring Programs Results. The result of each Breakout Group's input evaluation was analyzed using the Analytical Hierarchy Process through the Expert Choice software package (www.expertchoice.com). The results shown below for each Breakout group as well as the Aggregate results are taken directly from the summary output report from Expert Choice. These results define the weighted hierarchy for the application of the science objectives and criteria. The hierarchy assumed is the same for all Breakout groups and represents aggregate feedback from a group of six Maritimes Region managers compiled in 2003 following the Oceanstone meeting. The results discussed briefly below show the scaled "allocation" made to each Monitoring Program as a consequence of the information compiled from each Break Groups' importance grid feedback.

Breakout Group #1. This group favors two programs including (1) Groundfish Monitoring and (2) Crustacean and Invertebrates, which are calculated as having the same top priority. Ocean Science Monitoring followed closely by Hydrographic Monitoring are programs with the next highest priorities only slightly below those of the two highest priority ranking programs. Other programs fall significantly behind these top ranked programs. Fisheries Research Monitoring has the lowest priority of all the programs in the resultant priorities ranking.

Breakout Group #2. This group evaluates Toxic Chemical Monitoring with the top priority ranking followed closely by Hydrographic Monitoring. The closeness of the input evaluations results in priorities of other programs that are not much different.

Fisheries Research Monitoring again ranks as the lowest priority for Group #2 as for Group #1.

Breakout Group #3. This group favors Ocean Science Monitoring with the highest evaluated priority. Like evaluations for Crustacean and Invertebrates, Diadromous Fish Monitoring and Groundfish Monitoring result in tied second place rankings. The ranking for Hydrographic Monitoring compares closely and other programs all fall somewhat behind. Again, Fisheries Research Monitoring has the lowest priority of all programs.

Breakout Group #4. This group clearly favors Groundfish Monitoring in its evaluation for the highly weighted SFA and HaPAE tables. The result is that Groundfish Monitoring achieves the highest priorities in the rankings. Ocean Science Monitoring and Climate Change have identical evaluations resulting in the same second ranked priorities. Hydrographic Monitoring, Marine Ecosystem Productivity Total, Toxic Chemical Monitoring and Diadromous Fish Monitoring rank successively below. Other programs fall still further behind and different with other Groups, Env. Interact's with Marine Activity has the lowest ranked priority ranking.

Breakout Group #5. This group evaluates Hydrographic Monitoring at the highest possible level resulting in a higher priority than other programs. Fisheries Research Monitoring and Diadromous Fish Monitoring rank lowest. Other programs rank in the middle area with priorities that are comparable.

Aggregate Results. Overall, the results from the total evaluations of all breakout groups favor Groundfish Monitoring with the highest priority. Ocean Science Monitoring and Hydrographic Monitoring follow close behind. The remaining programs are clumped in priority with Fisheries Research Monitoring and Env. Interact's with Marine Activity with the lowest priority rankings.

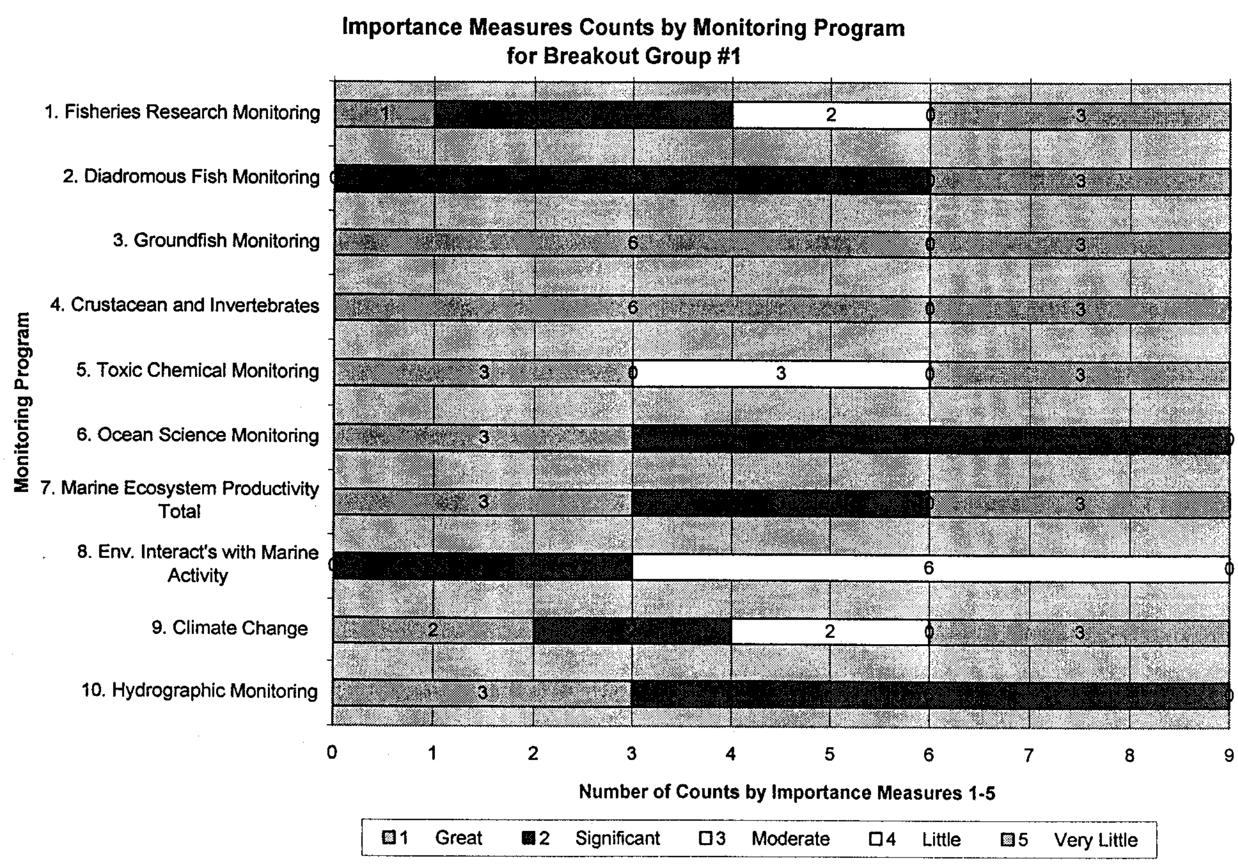
Breakout Group #1 Risk-based Prioritization Report

I. Input Data. Breakout Group #1 Monitoring Programs evaluation datasheet and summary.

Program Absolute Importance Evaluation relative to the Objective/Criteria combinations:
Scores Feedback Table – Enter values 1-5 in the table below corresponding to the importance of each monitoring program 1-10 below in contributing to each criteria/objective combination.

Program Importance Score:	*1*	*2*	*3*	*4*	*5*
Description	Great	Significant	Moderate	Little	Very Little

Objectives:	Sustainable Fish & Aquaculture			Healthy Ecosystems			Safe Waterways		
Criteria: / Programs:	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making
1. Fisheries Research Monitoring	3	2	1	3	2	2	5	5	5
2. Diadromous Fish Monitoring	2	2	2	2	2	2	5	5	5
3. Groundfish Monitoring	1	1	1	1	1	1	5	5	5
4. Crustacean and Invertebrates	1	1	1	1	1	1	5	5	5
5. Toxic Chemical Monitoring	3	3	3	1	1	1	5	5	5
6. Ocean Science Monitoring	2	2	2	1	1	1	2	2	2
7. Marine Ecosystem Productivity Total	2	2	2	1	1	1	5	5	5
8. Env. Interact's with Marine Activity	3	3	3	2	2	2	3	3	3
9. Climate Change	3	2	3	1	1	2	5	5	5
10. Hydrographic Monitoring	2	2	2	2	2	2	1	1	1



II. Model Output Results. Analytical Hierarchy Process (AHP) Results from Breakout Group #1 Monitoring Programs evaluation.

12/02/2004 11:29:48 PM

Page 1 of 6

Model Name: MonitoringProgs-BG#1

Treeview

- **Goal: Maritimes region monitoring program priority setting**
  - **Sustainable fisheries and aquaculture (L: .327)**
    - **Science excellence (L: .471)**
    - **Long-term efficiency (L: .221)**
    - **Decision making (L: .309)**
  - **Healthy and productive ecosystem (L: .547)**
    - **Science excellence (L: .420)**
    - **Long-term efficiency (L: .372)**
    - **Decision making (L: .208)**
  - **Safe and accessible waterways (L: .126)**
    - **Science excellence (L: .310)**
    - **Long-term efficiency (L: .226)**
    - **Decision making (L: .464)**

Alternatives

Fisheries Research Monitoring	.063
Diadromous Fish Monitoring	.069
Groundfish Monitoring	.135
Crustacean and Invertebrates	.135
Toxic Chemical Monitoring	.089
Ocean Science Monitoring	.130
Marine Ecosystem Productivity Total	.105
Env. Interact's with Marine Activity	.065
Climate Change	.085
Hydrographic Monitoring	.124

* *Distributive mode*

Dan Lane

Breakout Group #2 Risk-based Prioritization Report

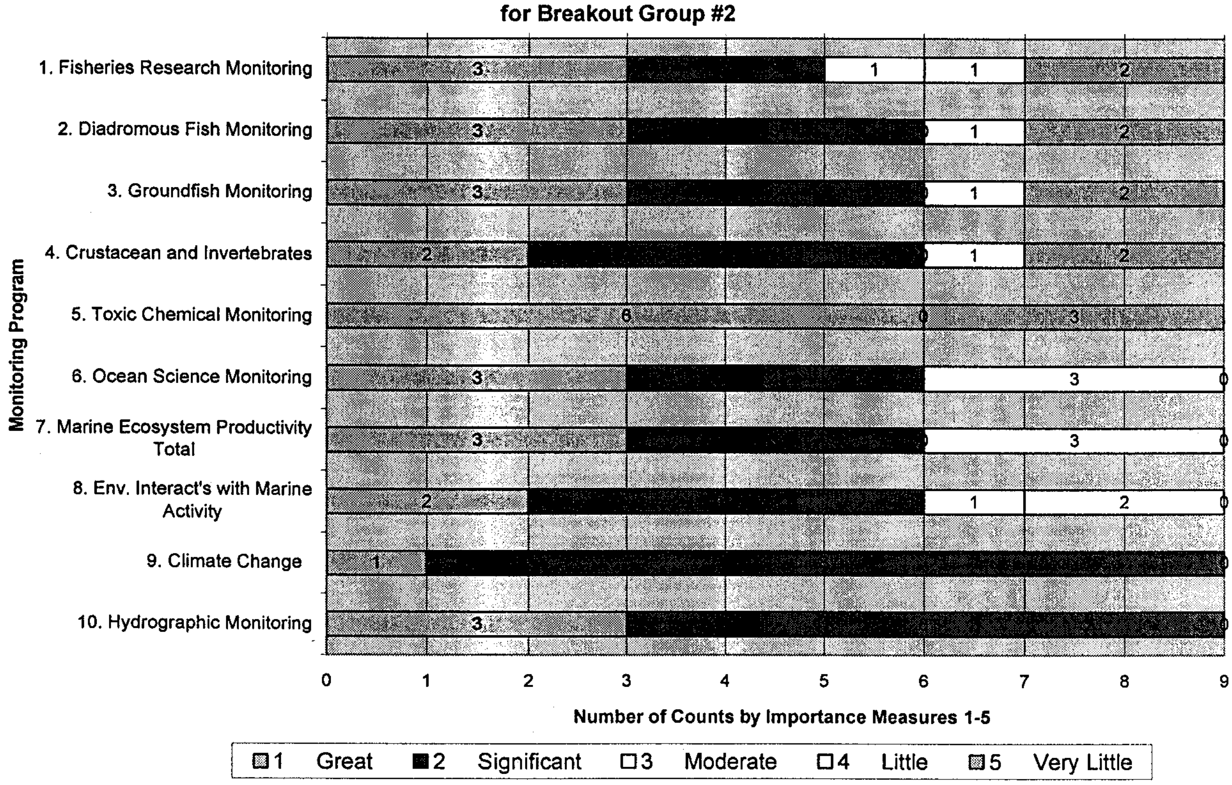
I. Input Data. Breakout Group #2 Monitoring Programs evaluation datasheet and summary.

Program Absolute Importance Evaluation relative to the Objective/Criteria combinations:
Scores Feedback Table – Enter values 1-5 in the table below corresponding to the importance of each monitoring program 1-10 below in contributing to each criteria/objective combination.

Program Importance Score:	*1*	*2*	*3*	*4*	*5*
Description	Great	Significant	Moderate	Little	Very Little

Objectives:	**Sustainable Fish & Aquaculture**			**Healthy Ecosystems**			**Safe Waterways**		
Criteria: / **Programs:**	**Science Excellence**	**Long-term Efficiency**	**Decision-Making**	**Science Excellence**	**Long-term Efficiency**	**Decision-Making**	**Science Excellence**	**Long-term Efficiency**	**Decision-Making**
1. Fisheries Research Monitoring	3	2	1	1	1	2	5	5	4
2. Diadromous Fish Monitoring	2	2	1	1	1	2	5	5	4
3. Groundfish Monitoring	2	2	1	1	1	2	5	5	4
4. Crustacean and Invertebrates	2	2	2	1	1	2	5	5	4
5. Toxic Chemical Monitoring	1	1	1	1	1	1	5	5	5
6. Ocean Science Monitoring	2	2	2	1	1	1	3	3	3
7. Marine Ecosystem Productivity Total	2	2	2	1	1	1	4	4	4
8. Env. Interact's with Marine Activity	2	2	1	1	2	2	4	4	3
9. Climate Change	2	2	2	2	1	2	2	2	2
10. Hydrographic Monitoring	2	2	2	2	2	2	1	1	1

Importance Measures Counts by Monitoring Program for Breakout Group #2



II. Model Output Results. Analytical Hierarchy Process (AHP) Results from Breakout Group #2 Monitoring Programs evaluation.

12/02/2004 11:38:05 PM

Page 1 of 7

Model Name: MonitoringProgs-BG#2

Treeview

- **Goal: Maritimes region monitoring program priority setting**
  - **Sustainable fisheries and aquaculture (L: .327)**
    - **Science excellence (L: .471)**
    - **Long-term efficiency (L: .221)**
    - **Decision making (L: .309)**
  - **Healthy and productive ecosystem (L: .547)**
    - **Science excellence (L: .420)**
    - **Long-term efficiency (L: .372)**
    - **Decision making (L: .208)**
  - **Safe and accessible waterways (L: .126)**
    - **Science excellence (L: .310)**
    - **Long-term efficiency (L: .226)**
    - **Decision making (L: .464)**

Alternatives

Fisheries Research Monitoring	.089
Diadromous Fish Monitoring	.096
Groundfish Monitoring	.096
Crustacean and Invertebrates	.089
Toxic Chemical Monitoring	.125
Ocean Science Monitoring	.106
Marine Ecosystem Productivity Total	.100
Env. Interact's with Marine Activity	.089
Climate Change	.097
Hydrographic Monitoring	.111

* *Distributive mode*

Dan Lane

Breakout Group #3 Risk-based Prioritization Report

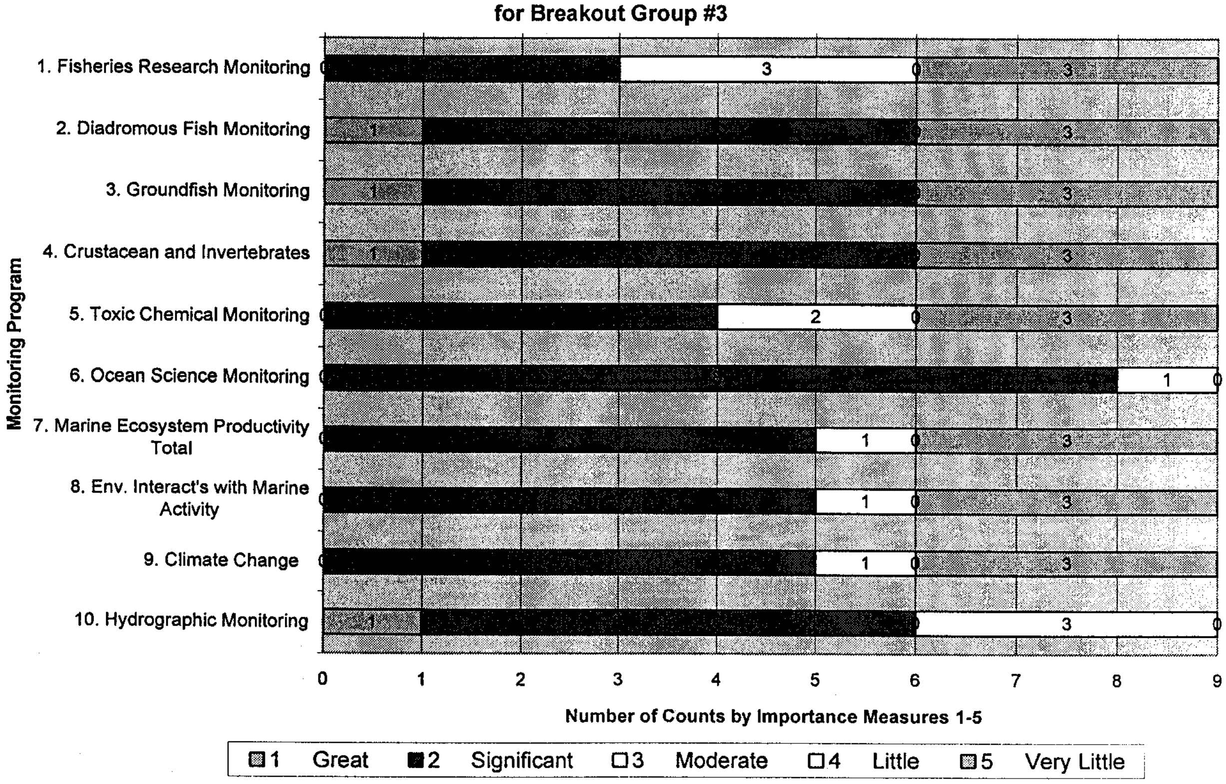
I. Input Data. Breakout Group #3 Monitoring Programs evaluation datasheet and summary.

Program Absolute Importance Evaluation relative to the Objective/Criteria combinations:
Scores Feedback Table – Enter values 1-5 in the table below corresponding to the importance of each monitoring program 1-10 below in contributing to each criteria/objective combination.

Program Importance Score:	*1*	*2*	*3*	*4*	*5*
Description	Great	Significant	Moderate	Little	Very Little

Objectives:	Sustainable Fish & Aquaculture			Healthy Ecosystems			Safe Waterways		
Programs: / Criteria:	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making
1. Fisheries Research Monitoring	3	2	3	3	2	2	5	5	5
2. Diadromous Fish Monitoring	2	2	1	2	2	2	5	5	5
3. Groundfish Monitoring	2	2	1	2	2	2	5	5	5
4. Crustacean and Invertebrates	2	2	1	2	2	2	5	5	5
5. Toxic Chemical Monitoring	3	3	2	2	2	2	5	5	5
6. Ocean Science Monitoring	2	2	3	2	2	2	2	2	2
7. Marine Ecosystem Productivity Total	2	2	3	2	2	2	5	5	5
8. Env. Interact's with Marine Activity	2	2	3	2	2	2	5	5	5
9. Climate Change	2	2	3	2	2	2	5	5	5
10. Hydrographic Monitoring	4	2	4	4	2	2	2	2	1

Importance Measures Counts by Monitoring Program for Breakout Group #3



II. Model Output Results. Analytical Hierarchy Process (AHP) Results from Breakout Group #3 Monitoring Programs evaluation.

12/02/2004 11:51:31 PM — Page 1 of 7

Model Name: MonitoringProgs-BG#3

Treeview

- **Goal: Maritimes region monitoring program priority setting**
  - **Sustainable fisheries and aquaculture (L: .327)**
    - **Science excellence (L: .471)**
    - **Long-term efficiency (L: .221)**
    - **Decision making (L: .309)**
  - **Healthy and productive ecosystem (L: .547)**
    - **Science excellence (L: .420)**
    - **Long-term efficiency (L: .372)**
    - **Decision making (L: .208)**
  - **Safe and accessible waterways (L: .126)**
    - **Science excellence (L: .310)**
    - **Long-term efficiency (L: .226)**
    - **Decision making (L: .464)**

Alternatives

Fisheries Research Monitoring	.071
Diadromous Fish Monitoring	.109
Groundfish Monitoring	.109
Crustacean and Invertebrates	.109
Toxic Chemical Monitoring	.086
Ocean Science Monitoring	.127
Marine Ecosystem Productivity Total	.094
Env. Interact's with Marine Activity	.094
Climate Change	.094
Hydrographic Monitoring	.106

* *Distributive mode*

Dan Lane

Breakout Group #4 Risk-based Prioritization Report

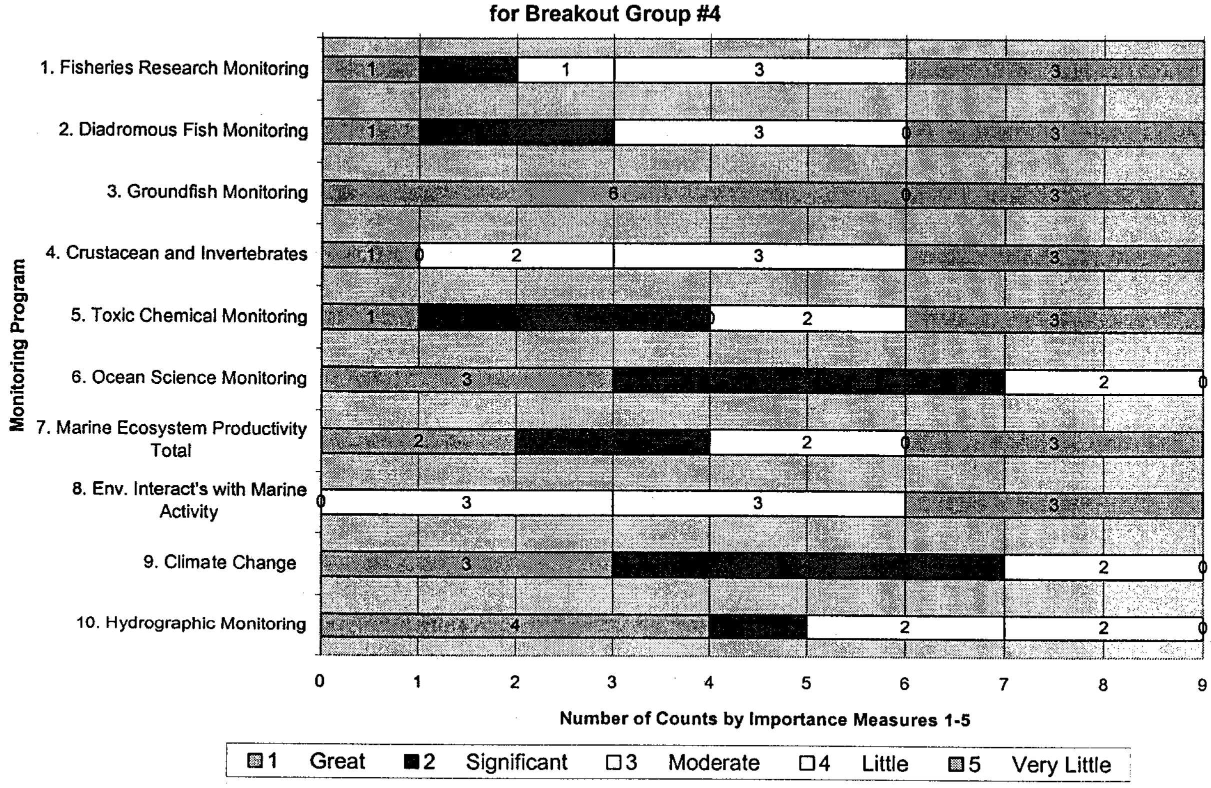
I. Input Data. Breakout Group #4 Monitoring Programs evaluation datasheet and summary.

Program Absolute Importance Evaluation relative to the Objective/Criteria combinations:
Scores Feedback Table – Enter values 1-5 in the table below corresponding to the importance of each monitoring program 1-10 below in contributing to each criteria/objective combination.

Program Importance Score:	1	2	3	4	5
Description	Great	Significant	Moderate	Little	Very Little

Objectives:	Sustainable Fish & Aquaculture			Healthy Ecosystems			Safe Waterways		
Criteria: / Programs:	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making
1. Fisheries Research Monitoring	3	2	1	4	4	4	5	5	5
2. Diadromous Fish Monitoring	3	3	1	2	3	2	5	5	5
3. Groundfish Monitoring	1	1	1	1	1	1	5	5	5
4. Crustacean and Invertebrates	3	3	1	4	4	4	5	5	5
5. Toxic Chemical Monitoring	4	4	2	2	1	2	5	5	5
6. Ocean Science Monitoring	1	2	3	2	1	3	2	2	1
7. Marine Ecosystem Productivity Total	1	2	3	2	1	3	5	5	5
8. Env. Interact's with Marine Activity	4	4	4	3	3	3	5	5	5
9. Climate Change	1	2	3	2	1	3	2	2	1
10. Hydrographic Monitoring	4	4	3	2	1	3	1	1	1

Importance Measures Counts by Monitoring Program for Breakout Group #4



II. Model Output Results. Analytical Hierarchy Process (AHP) Results from Breakout Group #4 Monitoring Programs evaluation.

12/02/2004 11:52:40 PM

Page 1 of 7

Model Name: MonitoringProgs-BG#4

Treeview

- **Goal: Maritimes region monitoring program priority setting**
  - **Sustainable fisheries and aquaculture (L: .327)**
    - **Science excellence (L: .471)**
    - **Long-term efficiency (L: .221)**
    - **Decision making (L: .309)**
  - **Healthy and productive ecosystem (L: .547)**
    - **Science excellence (L: .420)**
    - **Long-term efficiency (L: .372)**
    - **Decision making (L: .208)**
  - **Safe and accessible waterways (L: .126)**
    - **Science excellence (L: .310)**
    - **Long-term efficiency (L: .226)**
    - **Decision making (L: .464)**

Alternatives

Fisheries Research Monitoring	.052
Diadromous Fish Monitoring	.082
Groundfish Monitoring	.183
Crustacean and Invertebrates	.047
Toxic Chemical Monitoring	.090
Ocean Science Monitoring	.138
Marine Ecosystem Productivity Total	.110
Env. Interact's with Marine Activity	.040
Climate Change	.138
Hydrographic Monitoring	.119

* *Distributive mode*

Dan Lane

Breakout Group #5 Risk-based Prioritization Report

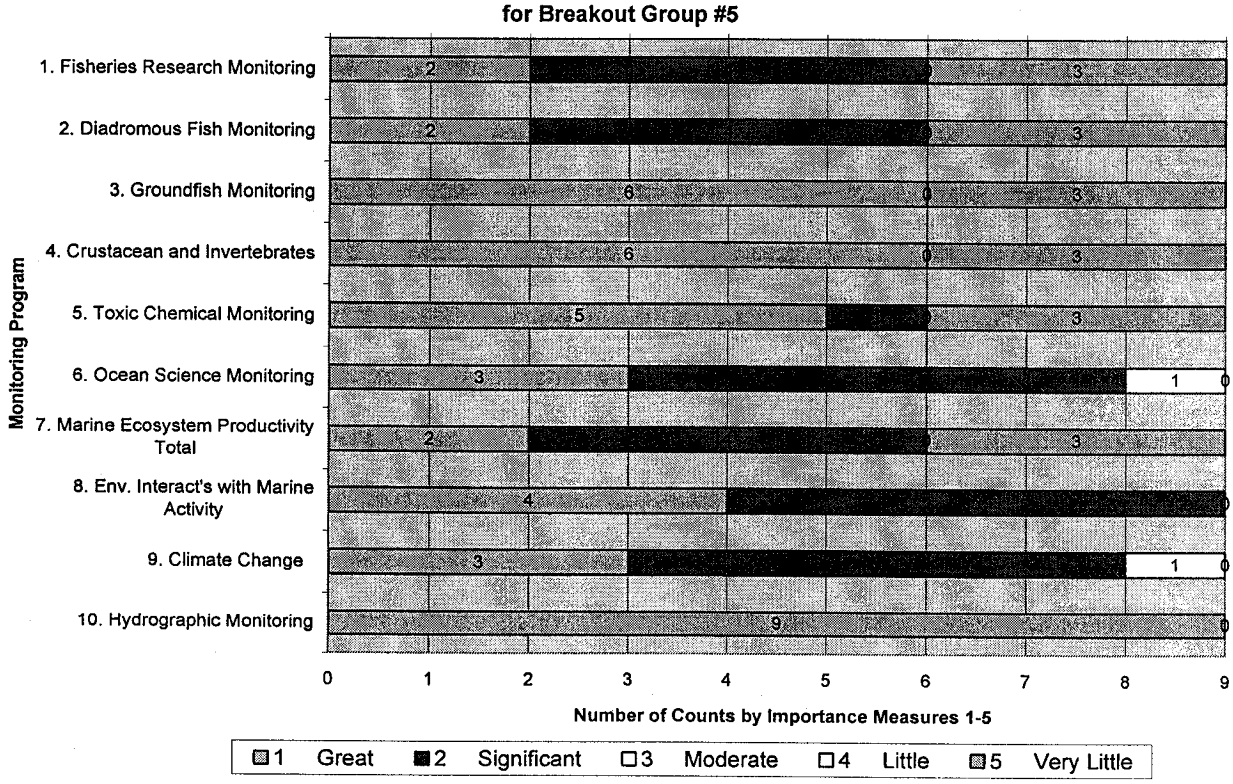
I. Input Data. Breakout Group #5 Monitoring Programs evaluation datasheet and summary.

Program Absolute Importance Evaluation relative to the Objective/Criteria combinations:
Scores Feedback Table – Enter values 1-5 in the table below corresponding to the importance of each monitoring program 1-10 below in contributing to each criteria/objective combination.

Program Importance Score:	*1*	*2*	*3*	*4*	*5*
Description	Great	Significant	Moderate	Little	Very Little

Objectives:	**Sustainable Fish & Aquaculture**			**Healthy Ecosystems**			**Safe Waterways**		
Criteria: / **Programs:**	**Science Excellence**	**Long-term Efficiency**	**Decision-Making**	**Science Excellence**	**Long-term Efficiency**	**Decision-Making**	**Science Excellence**	**Long-term Efficiency**	**Decision-Making**
1. Fisheries Research Monitoring	2	1	1	2	2	2	5	5	5
2. Diadromous Fish Monitoring	2	1	1	2	2	2	5	5	5
3. Groundfish Monitoring	1	1	1	1	1	1	5	5	5
4. Crustacean and Invertebrates	1	1	1	1	1	1	5	5	5
5. Toxic Chemical Monitoring	1	1	2	1	1	1	5	5	5
6. Ocean Science Monitoring	1	2	3	1	2	2	1	2	2
7. Marine Ecosystem Productivity Total	1	2	2	1	2	2	5	5	5
8. Env. Interact's with Marine Activity	1	2	1	1	2	1	2	2	2
9. Climate Change	1	2	3	1	2	2	1	2	2
10. Hydrographic Monitoring	1	1	1	1	1	1	1	1	1

Importance Measures Counts by Monitoring Program for Breakout Group #5



II. Model Output Results. Analytical Hierarchy Process (AHP) Results from Breakout Group #5 Monitoring Programs evaluation.

12/02/2004 11:57:48 PM

Page 1 of 7

Model Name: MonitoringProgs-BG#5

Treeview

- **Goal: Maritimes region monitoring program priority setting**
  - **Sustainable fisheries and aquaculture (L: .327)**
    - **Science excellence (L: .471)**
    - **Long-term efficiency (L: .221)**
    - **Decision making (L: .309)**
  - **Healthy and productive ecosystem (L: .547)**
    - **Science excellence (L: .420)**
    - **Long-term efficiency (L: .372)**
    - **Decision making (L: .208)**
  - **Safe and accessible waterways (L: .126)**
    - **Science excellence (L: .310)**
    - **Long-term efficiency (L: .226)**
    - **Decision making (L: .464)**

Alternatives

Fisheries Research Monitoring	.068
Diadromous Fish Monitoring	.068
Groundfish Monitoring	.112
Crustacean and Invertebrates	.112
Toxic Chemical Monitoring	.105
Ocean Science Monitoring	.098
Marine Ecosystem Productivity Total	.078
Env. Interact's with Marine Activity	.111
Climate Change	.098
Hydrographic Monitoring	.150

* *Distributive mode*

Dan Lane

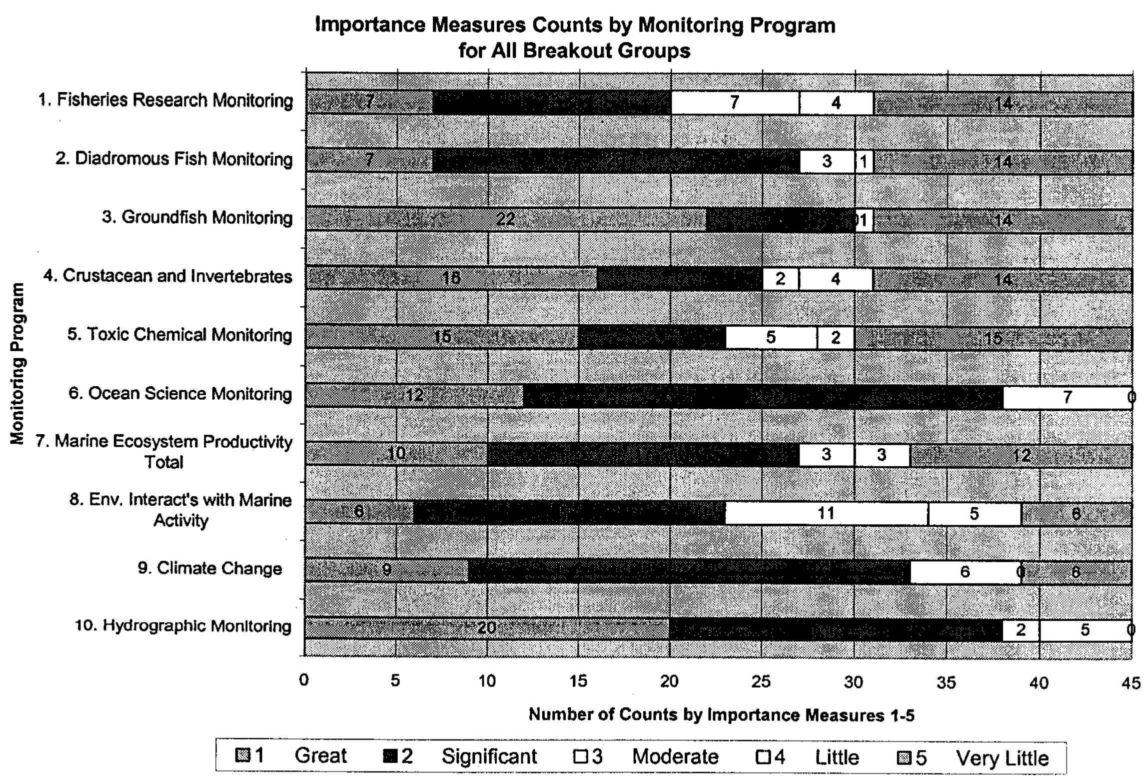
Aggregated Breakout Group Risk-based Prioritization Report

I. Input Data. Aggregated Breakout Group Monitoring Programs evaluation datasheet and summary. Values are averaged by cell over all 5 Breakout Groups' responses.

Program Absolute Importance Evaluation relative to the Objective/Criteria combinations:
Scores Feedback Table – Enter values 1-5 in the table below corresponding to the importance of each monitoring program 1-10 below in contributing to each criteria/objective combination.

Program Importance Score:	*1*	*2*	*3*	*4*	*5*
Description	Great	Significant	Moderate	Little	Very Little

Objectives:	Sustainable Fish & Aquaculture			Healthy Ecosystems			Safe Waterways		
Criteria: / Programs:	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making	Science Excellence	Long-term Efficiency	Decision-Making
1. Fisheries Research Monitoring	2.80	1.80	1.40	2.60	2.20	2.40	5.00	5.00	4.80
2. Diadromous Fish Monitoring	2.20	2.00	1.20	1.80	2.00	2.00	5.00	5.00	4.80
3. Groundfish Monitoring	1.40	1.40	1.00	1.20	1.20	1.40	5.00	5.00	4.80
4. Crustacean and Invertebrates	1.80	1.80	1.20	1.80	1.80	2.00	5.00	5.00	4.80
5. Toxic Chemical Monitoring	2.40	2.40	2.00	1.40	1.20	1.40	5.00	5.00	5.00
6. Ocean Science Monitoring	1.60	2.00	2.60	1.40	1.40	1.80	2.00	2.20	2.00
7. Marine Ecosystem Productivity Total	1.60	2.00	2.40	1.40	1.40	1.80	4.80	4.80	4.80
8. Env. Interact's with Marine Activity	2.40	2.60	2.40	1.80	2.20	2.00	3.80	3.80	3.60
9. Climate Change	1.80	2.00	2.80	1.60	1.40	2.20	3.00	3.20	3.00
10. Hydrographic Monitoring	2.60	2.20	2.40	2.20	1.60	2.00	1.20	1.20	1.00



II. Model Output Results. Analytical Hierarchy Process (AHP) Results from aggregated Breakout Group Monitoring Programs evaluation.

Model Name: MonitoringProgs-Aggregated

Treeview

- **Goal: Maritimes region monitoring program priority setting**
  - **Sustainable fisheries and aquaculture (L: .327)**
    - **Science excellence (L: .471)**
    - **Long-term efficiency (L: .221)**
    - **Decision making (L: .309)**
  - **Healthy and productive ecosystem (L: .547)**
    - **Science excellence (L: .420)**
    - **Long-term efficiency (L: .372)**
    - **Decision making (L: .208)**
  - **Safe and accessible waterways (L: .126)**
    - **Science excellence (L: .310)**
    - **Long-term efficiency (L: .226)**
    - **Decision making (L: .464)**

Alternatives

Fisheries Research Monitoring	.067
Diadromous Fish Monitoring	.086
Groundfish Monitoring	.130
Crustacean and Invertebrates	.093
Toxic Chemical Monitoring	.100
Ocean Science Monitoring	.124
Marine Ecosystem Productivity Total	.101
Env. Interact's with Marine Activity	.076
Climate Change	.100
Hydrographic Monitoring	.123

* *Distributive mode*

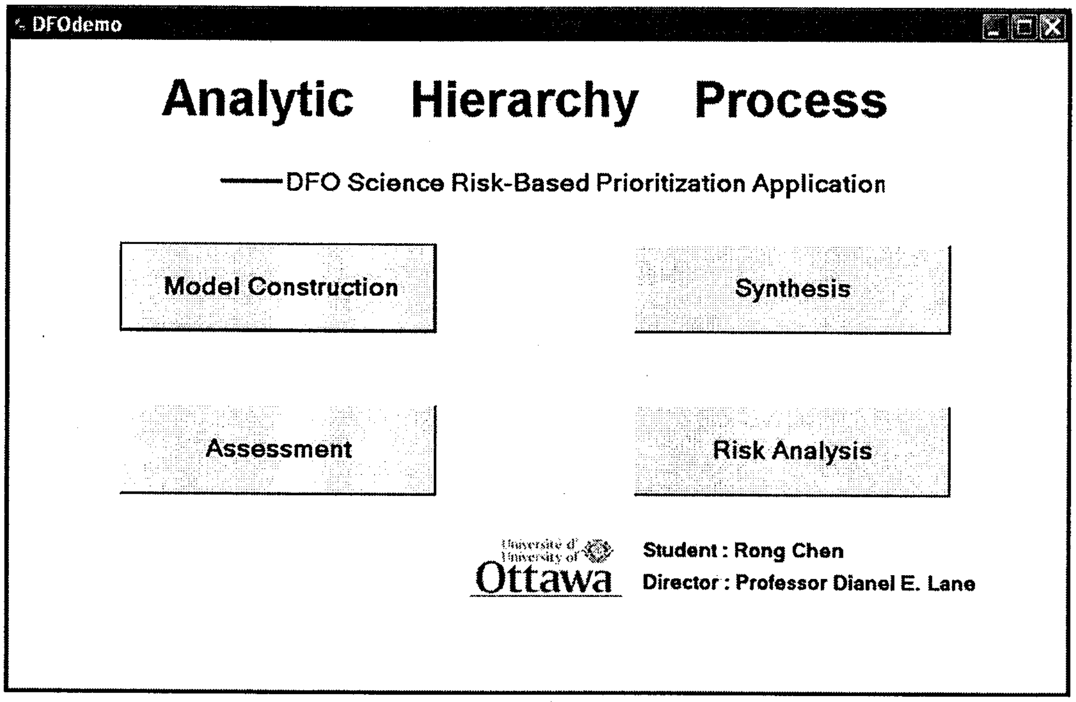
Dan Lane

Appendix IV.

Application System Help

- Create new model

Figure IV-1: Main screen



1. From main screen (Figure IV-1) , select "Model Construction"
2. From menu "Model", select "New" to create a new model
3. Type the goal of model and select its formulation in new model creation screen (Figure IV-2), then click "OK"
4. Click the top node of hierarchy just created, from menu "Node", click "insert child" to input the objective under the goal of hierarchy
5. Repeat step 4 to insert other objective, or click the objective just created, from menu "Node", click "Insert Sibling" to insert other objectives.
6. Use the sub menu under "Node" shown in Figure IV-3 to insert, delete, edit, copy and paste the objectives and criteria within the hierarchy
7. From menu "Alternative", click "Add" to add an alternative, or select an existing alternative, then click "Delete" to delete it.

8. After finish the model, from menu "Participant", click "Setup" to set the participant for the model as shown in Figure 4-4, or use "Edit" to edit the information for participants (Figure 4-3) and then set up
9. Close the model construction module and go back to main screen

Figure IV-2: Create new model

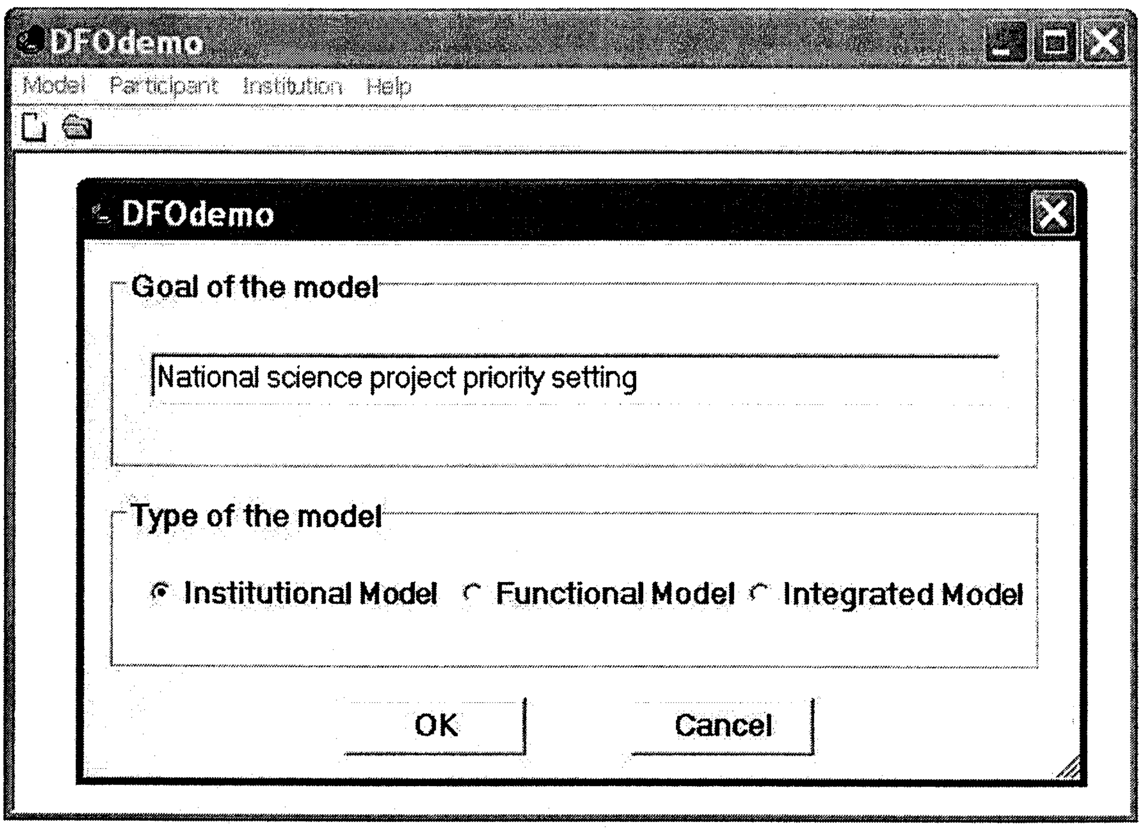
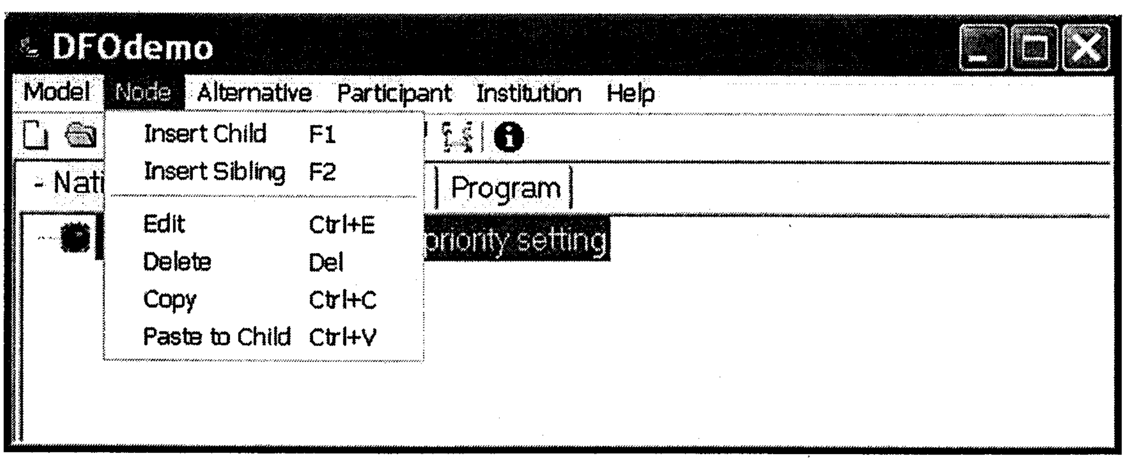


Figure IV-3: Edit element of hierarchy



- Assessment process of the model

1. From main screen (Figure IV-1), click "Assessment"
2. Select the model needed to be assessed
3. Select the goal of hierarchy
4. Click "Verbal pairwise comparison" tab or "Graphical pairwise comparison" tab
5. Select the participant making the judgment, click "OK"
6. Click and input data into the cell in datagrid "Pairwise comparison", or select the cell, and then move the track bar in "Verbal pairwise comparison" tab (Figure 4-6) or the slider/pie chart in "Graphical pairwise comparison" tab (Figure 4-7) to make the judgment
7. After finish the judgment, click the "Model View" tab to go back to the model
8. Select the objectives under the goal respectively and repeat step 4, 6, and 7 to finish the pairwise comparison with respect to all objectives
9. After finish all the pairwise comparison, from menu "Pairwise", click "Calculate" to calculate the local priorities of objectives and criteria
10. Click "Datagrid input" tab to begin datagrid evaluation for alternatives (Figure 4-9)
11. Click button "Rating Setup" to go to rating setup sub-module to set the rating scale for evaluation first (Figure 4-8)
12. Select the existing rating scale or create a new one, then click "Select"
13. To create a new scale, input the name of rate in the datagid, after finish, click "Assess" to do pairwise comparison between rates, then click "Calculate" to calculate corresponding priorities, then close the module and save the scale
14. To use an existing, simply close the module and save the scale
15. Input the evaluation into datagrid by input the number on top of the different rates into datagrid "Scale Value", the corresponding priorities of each rate is then saved in the datagrid "Absolute Value"
16. After finish all the evaluation, click "Calculate" from menu "Datagrid" to calculate the global priorities for alternatives
17. Save the changes
18. Close the module and go back to main screen

- Synthesis results
  1. From main screen (Figure IV-1), click "Synthesis"
  2. Select the model with the assessment done
  3. The resulted priorities is shown following each item in the hierarchy in "Model View" tab (Figure 4-11)
  4. Click "Priorities ranking" tab to view the ranking bar for alternatives' resulted priorities ranking (Figure 4-12)
  5. Click "Actual Allocation" tab to view the budget allocation pie chart
  6. Input the total dollar value which can be allocated, then click "Show" to see the resulted pie chart
  7. Click "Save" to save the allocation results for next level model allocation
  8. Close the module and go back to main screen

- Risk Analysis
  1. From main screen (Figure IV-1), click "Risk Analysis"
  2. Select the model needed to do the risk analysis
  3. Click "Objective Dynamic" tab to do the top-down risk analysis (Figure 4-14)
  4. Move the slider under one of the objective to change the weight of it
  5. Observe the resulted priority changes for alternatives
  6. Click "Performance Dynamic" tab to do the bottom-up risk analysis (Figure 4-15)
  7. Choose the alternative from combo box
  8. Move the slider under the alternative to change the datagrid value of it
  9. Observe the corresponding changes in alternatives' priorities
  10. Click "Utility Analysis" tab to do the utility analysis (Figure 4-16)
  11. Change the thresholds of alternative in "Alternative Thresholds" datagrid, or "Current Total Budget" value
  12. Observe the corresponding changes in total utility
  13. Close the module and go back to the main screen

Appendix V.

Maritimes Projects Aggregation

This appendix document is a spreadsheet file in Excel format generated based on another Excel file provided by Sherry Niven from DFO science Maritimes Region, "Projects Maritimes - by Functions & Tables (Strategic Outcomes) (July 22)" (DFO 2003d). This document identifies some 400 regional science programs and their expected contributions to DFO objectives and functions. The file organizes the Maritimes Region science projects by DFO strategic outcomes and six science functions. Furthermore, the data sheet characterizes individual science project numerically by different dollar values assigned to it and the percentage of its contribution toward every Function and Table.

Fourth-Level Business Line and their Respective Projects	Grand Total ($K)	Monitoring / Surveillance	Data Management / Gestion de données	Advice / Conseil	Strategic Research / Recherche stratégique	Products/Services / Produits/Services	Management / Gestion	Table 1. Sustainable Fisheries and Aquaculture	Table 2. Healthy and Productive Aquatic Ecosystems	Table 3. Safe and Accessible Waterways
Fisheries Research, General - 1110										
Biodiversity Studies	38	-	-	19	19	-	-	25	75	
CDEENA (Comparative Dynamics of Exploited Ecosystems in the Northwest Atlantic)	364	-	-	-	364	-	-		100	
Data Visualization and Management	56	-	56	-	-	-	-	50	50	
Fisheries Systems Evaluation	64	-	-	-	64	-	-	100		
Otolith Studies	68	14	-	34	20	-	-	100		
Migration Studies	28	-	-	14	14	-	-	100		
National Sampling Program	205	205	-	-	-	-	-	100		
Oceanographic Data Handling (ODH)	182	145	36	-	-	-	-	40	40	20
Stock Assessment Methods	79	-	8	55	16	-	-	80	20	
Virtual Data Centre (BIO)	-	-	-	-	-	-	-	50	50	
Gulf of Maine and Pelagic Fisheries - Program Direction	98	-	-	-	-	-	98	75	25	
Scotian Shelf Program Direction	92	-	-	-	-	-	92	70	30	
Marine Fish Division Administration	424	-	-	-	-	-	424	60	40	
Species at Risk	115	-	-	6	109	-	-	-	100	-
International Observer Program	130	117	13	-	-	-	-	80	20	
Fisheries Research, General - 1110 Total	**1,944**	**481**	**113**	**137**	**597**	**-**	**616**	**53**	**45**	**2**
Groundfish - 1112										
Fisheries Assessments and Assoc. Research	1,078.7	115	-	856	108	-	-	98	2	-
Cod Stock Migration and Mixing Patterns Within and Among Mgnt Units in the Gulf of Maine Region	68	-	-	-	68	-	-	100		
Groundfish Age Determinations	74.7	-	-	63	11	-	-	100		
Groundfish Management Research	137	-	-	123	14	-	-	100		
Groundfish Trawl Surveys	904	633	-	271	-	-	-	40	60	
Offshore Development	36	-	4	32	-	-	-		100	
Sentinel Survey - Scotian Shelf	500	400	-	100	-	-	-	100		
Groundfish - 1112 Total	**2,797**	**1,160**	**4**	**1,430**	**203**	**-**	**-**	**79**	**21**	**-**
Pelagic Fish - 1113										
Pelagics Assessment and Assoc. Research	348.8	70	-	227	52	-	-	100	-	-
Hydroacoustics	192	19	-	153	19	-	-	100		
Sharks	101	10	-	60	30	-	-	80	20	
Pelagic Fish - 1113 Total	**641**	**99**	**-**	**430**	**112**	**-**	**-**	**97**	**3**	**-**
Marine Mammals - 1115										
Discovery Channel Project	100	-	-	-	100	-	-		100	
General	2	-	-	-	-	-	2		100	
Grey Seal Ecology - Sable Island	41	-	-	4	36	-	-	10	90	
Marine Mammal Studies	173	-	-	26	147	-	-	-	100	-
Right Whale	68	-	-	-	68	-	-		100	
Marine Mammals - 1115 Total	**383**	**-**	**-**	**18**	**363**	**-**	**2**	**1**	**99**	**-**
Species at Risk - 1132										
Leatherback Turtle	179	-	-	-	179	-	-		100	
Right Whale	309	-	-	-	309	-	-		100	
Species at Risk - 1132 Total	**488**	**-**	**-**	**-**	**488**	**-**	**-**	**-**	**100**	**-**
Marine Fish, Total	**6,253**	**1,740**	**117**	**2,015**	**1,764**	**-**	**617**	**62**	**38**	**1**
Diadromous and Freshwater Fish - 1111								-		
Aboriginal Fishing Rights (Supreme Court)	23	-	-	-	-	23	-	100		
Acadia First Nation Sponsorship Training Agreement	3	-	-	3	-	-	-	100		
Atlantic Salmon Northern Maine	5	-	-	-	-	5	-	100		
Coordination and Outreach	69	-	-	69	-	-	-	50	50	
DFO-NSPC Partnering Agreement	17	9	-	-	9	-	-	100		
Diadromous Fish Division Coordination	193	-	-	19	-	19	154	10	90	
Diadromous Statistics	65	-	32	-	32	-	-	10	90	
Division Management - Diadromous	153	-	-	-	-	-	153	10	90	
Division Special Projects	176	-	-	-	176	-	-	10	90	
Engineering	155	-	-	78	-	78	-	70	30	
IRF IBOF 3-Rivers Monitoring	30	30	-	-	-	-	-	10	90	
Kingsclear - Living Gene Bank	100	-	-	-	-	100	-	10	90	
Kingsclear Fisheries Conserv. & Dev.	58	-	-	-	-	58	-	10	90	

Mactaquac Fish Collection Facility	112	100	-	11	-	-	-	100		
Mactaquac Hatchery Operations	446	-	-	-	89	357	-	10	90	
Maritimes Non-Salmon	144	72	-	14	58	-	-	100		
Nova Scotia and Inner Bay of Fundy Salmon	263	132	-	26	105	-	-	10	90	
Nova Scotia Biodiversity Units	203	-	-	-	41	163	-	10	90	
South and Western N.B. Salmon	208	104	-	21	83	-	-	10	90	
Diadromous and Freshwater Fish - 1111 Total	**2,422**	**447**	**32**	**241**	**593**	**802**	**307**	**26**	**74**	**-**
Species at Risk - 1132										
Atlantic Whitefish	191	-	-	19	96	76	-		100	
Inner Bay of Fundy Salmon	61	-	-	6	18	37	-		100	
Inner Bay of Fundy Salmon	130	-	-	13	26	91	-		100	
Inner Bay of Fundy Salmon	85	34	-	9	43	-	-		100	
Inner Bay of Fundy Salmon	92	-	-	9	82	-	-		100	
Inner Bay of Fundy Salmon	368	37	-	37	175	120	-	-	100	-
Lake Utopia Dwarf Smelt	17	-	-	-	17	-	-		100	
Species at Risk - 1132 Total	**575**	**34**	**-**	**56**	**281**	**204**	**-**	**-**	**100**	**-**
Diadromous Fish, Total	**2,998**	**481**	**32**	**297**	**874**	**1,007**	**307**	**21**	**79**	**-**
Invertebrates and Other Species - 1114										
Shrimp Assessment and Research	225	22	11	146	45	-	-	100	-	-
Crustacean Projects	706	141	59	306	141	-	59	97	3	-
Non-Crustacean Projects	40	-	-	20	20	-	-	100		
Coastal Ecosystems Issues	66	-	-	33	33	-	-	-	100	
Scallop Projects	1,333	347	160	453	373	-	-	92	8	-
DFO Native Internship Program	45	-	-	45	-	-	-	100		
Division Administration	213	-	-	-	-	-	213	80	20	-
First Nations	12	-	-	12	-	-	-	100		
Full Bay Scallop Association FY 02-03	7	4	-	-	4	-	-	100		
Inshore Molluscs	17	-	-	15	2	-	-	100		
Invertebrate Biology Section	7	-	-	-	-	-	7	50	50	
Molluscan Hatchery	24	-	-	-	24	-	-	50	50	
Offshore Clam Research Project	109	33	11	44	22	-	-	100	-	-
Oyster Research	71	-	-	36	36	-	-	62	38	-
Sea Cucumber 4X/5Z	25	13	-	13	-	-	-	100		
Sea Urchin in LFAs 36 and 38	19	4	-	13	2	-	-	100		
Section Activities	66	-	17	17	-	-	33	100		
Shellfish Biology	18	-	-	-	18	-	-	50	50	
Survey Design Research	102	-	-	-	102	-	-	100		
Invertebrates and Other Species - 1114 Total	**3,106**	**661**	**237**	**1,107**	**824**	**-**	**277**	**91**	**9**	**-**
Marine Plants - 1116										
Marine Plants	141	42	-	64	35	-	-	100		
Marine Plants - 1116 Total	**141**	**42**	**-**	**64**	**35**	**-**	**-**	**100**	**-**	**-**
Biodiversity Science - 1131										
Biodiversity Genomics	253	-	-	-	253	-	-		100	
Centre for Marine Biodiversity	336	-	-	336	-	-	-		100	
Triploid Scallops - Maritimes	8	-	-	-	8	-	-		100	
Biodiversity Science - 1131 Total	**597**	**-**	**-**	**336**	**261**	**-**	**-**	**-**	**100**	**-**
Species at Risk - 1132										
Atlantic Whitefish	49	-	-	-	49	-	-		100	
Atlantic Whitefish-McPherson	-	-	-	-	-	-	-		100	
Atlantic Whitefish	49	-	-	-	49	-	-	-	100	-
Harbour Porpoise	10	-	-	-	10	-	-		100	
Inner Bay of Fundy Salmon	52	-	-	10	41	-	-		100	
Lake Utopia Dwarf Smelt	-	-	-	-	-	-	-		100	
Leatherback Turtle	-	-	-	-	-	-	-		100	
Right Whale	65	-	-	-	65	-	-		100	
SARP Coordination	451	-	-	-	-	-	451		100	
Scientific Core Capacity	65	-	-	-	65	-	-		100	
Species at Risk - 1132 Total	**691**	**-**	**-**	**10**	**230**	**-**	**451**	**-**	**100**	**-**
Invertebrates Fisheries, Total	**4,536**	**704**	**237**	**1,518**	**1,350**	**-**	**728**	**65**	**35**	**-**
Environmental Science, General - 1120										
Adm.-(ESSRF) Environmental Science Strategic Research Fund	-	-	-	-	-	-	-		100	
Division Administration and Management	174	-	-	-	-	-	174		100	
Environmental Advice	38	-	-	35	-	-	3	-	100	-
General	65	-	-	65	-	-	-		100	
Environmental Science, General - 1120 Total	**277**	**-**	**-**	**98**	**-**	**-**	**179**	**-**	**100**	**-**
Toxic Chemicals - 1121										
Assessment of Natural and Arthropogenic Impacts of Contaminants	173	-	9	17	147	-	-		100	
Biochemical Indicators of the Health of Aquatic Animals	67	-	3	7	57	-	-		100	

Biological Impacts of Sewage Contaminants Discharged to Halifax Harbour	322	-	16	32	273	-	-	-	100	-
Clay-Oil Flocculation	10	5	-	-	5	-	-	-	100	-
Deep Water Benthic Community Study	387	-	-	39	348	-	-		100	
Ecotoxicity of Organic Contaminants	197	-	-	20	177	-	-		100	
Fate & Effects of Chemically Dispersants/Oil/Producted Water	630	-	-	63	567	-	-	-	100	-
Indiana Harbour Canal Shoreline	182	-	-	18	164	-	-		100	
ONR/Ohio State University	75	-	-	7	67	-	-		100	
ONR-Strataform	213	-	-	21	191	-	-		100	
Point Lepreau Environmental Monitoring Program	273	260	14	-	-	-	-		100	
Prevalence of Methylmercury...Bay of Fundy	165	-	8	16	140	-	-		100	
Project with University of Cincinnati	35	-	-	-	35	-	-		100	
Toxicity Identification & Evaluation (TIE of Toxic Components in Residual Oil)	118	-	-	-	118	-	-		100	
Trace Metal Transport into the Western North Atlantic	84	-	4	-	80	-	-		100	
Toxic Chemicals - 1121 Total	**2,929**	**265**	**54**	**241**	**2,369**	**-**	**-**	**-**	**100**	**-**
Fish Habitat - 1123										
Aquatic Toxicology of Marine Phytotoxins	82	-	4	12	65	-	-	50	50	
Benthic Ecology Research	71	-	-	7	63	-	-		100	
Coastal Habitat Studies	55	-	-	6	50	-	-		100	
Effects of Change in Coastal and Marine Environments ...	510	-	-	-	510	-	-		100	
Estuarine Nutrient Dynamics	76	-	4	4	68	-	-		100	
Harmful Marine Algal Research	131	-	3	10	118	-	-	50	50	-
Risk Assessment of Organic Chemistry to Fisheries	61	-	-	6	55	-	-		100	
Sedimentology of Coastal Habitats	35	-	-	3	31	-	-		100	
Spatial Utilization of Benthic Habitat by Demersal Fish	747	-	19	37	672	19	-	-	100	-
Fish Habitat - 1123 Total	**1,767**	**-**	**33**	**95**	**1,621**	**18**	**-**	**6**	**94**	**-**
Environmental Effects of Aquaculture - 1124										
Antibiotic Residues from Salmon Farms in Southwest NB	112	-	-	11	101	-	-	50	50	
Aquaculture	75	-	-	5	70	-	-	50	50	-
Aquaculture Ecology Research	79	-	-	8	71	-	-	50	50	
Endocrine Disruption in Invertebrates...American Lobster	149	-	-	15	134	-	-	50	50	
Environmental Studies for Sustainable Aquaculture	333	-	17	33	283	-	-	50	50	-
Integrated Ecosystem Studies for Modelling Mussel Aquacult.-Environ. Interact.	441	-	11	44	386	-	-	50	50	-
Environmental Effects of Aquaculture - 1124 Total	**1,189**	**-**	**39**	**117**	**1,033**	**-**	**-**	**50**	**50**	**-**
Environmental Data & Advice - 1125										
Assessment of Natural and Arthropogenic Impacts of Contaminants	19	-	-	19	-	-	-	10	90	
Atlas of Fish Spawning, Nursery & Larval Concentration Areas on the Scotian Shelf	95	-	-	95	-	-	-	10	90	
Data Management	66	-	66	-	-	-	-	10	90	
Division Administration and Management	43	-	-	43	-	-	-	10	90	
Environmental Advice	130	-	-	130	-	-	-	10	90	-
Sedimentology of Coastal Habitats	4	-	-	4	-	-	-	10	90	
Environmental Data & Advice - 1125 Total	**357**	**-**	**66**	**292**	**-**	**-**	**-**	**10**	**90**	**-**
Environmental Science, Total	**6,518**	**265**	**192**	**843**	**5,022**	**18**	**179**	**11**	**89**	**-**
Oceans Science, General - 1210										
Atlantic Shelf Model (Currents and Circulation - Atlantic Offshore)	304	91	-	30	182	-	-	33	34	33
Biogeochemical and Bio-optical Modelling of Carbon Cycling and of Air-Sea CO2 Fluxes	152	-	-	-	152	-	-		100	
Biological and Physical Processes Controlling Basin-Scale Productivity in the Northwest Atlantic	174	-	-	-	174	-	-		100	
Brooke Ocean Technology	31	9	-	-	6	16	-		100	
Calanus finmarchicus - the NW Atlantic Ecosystem	40	-	-	-	40	-	-		100	
Circulation and Interannual Variability in the Newfoundland Basin	62	-	-	-	62	-	-	15	60	25
Continuous Plankton Recorder (CPR) Sampling Program	192	192	-	-	-	-	-	50	50	
Developing an XML Application for DFO Marine Data Exchange	20	-	20	-	-	-	-	33	34	33
Development of a Mixing Rate Climatology for the Scotian Shelf	137	-	-	69	69	-	-	30	70	
Divisional Administration	1,069	-	-	-	-	-	1,069	3	50	47
East Coast Climate Change (Sea-Ice Conditions - Eastern Canada Climate Change)	252	25	-	101	126	-	-		20	80
Effects of Foodweb Struct. and Phys. Variability on the Partitioning of Biogenic Carbon and Efficiency of its Vertical Transport in the Labrador Sea	65	-	-	7	59	-	-		100	
Environmental Model for Drilling Waste	153	46	-	61	46	-	-	30	70	
Environmental Studies for Sustainable Aquaculture	-	-	-	-	-	-	-	70	30	
Fate and Effects of Produced Water	-	-	-	-	-	-	-	30	70	

General	100	-	-	-	-	-	100	46	49	6
Incorporate Models	10	-	-	-	10	-	-		100	
Integrated Ecosystem Studies for Modelling Mussel Aquacult.-Environ. Interact.		-	-	-	-	-	-	70	30	
Interannual Variability of the North Atlantic Meridional Overturning Circulation as Observed in the Labrador Sea	445	-	-	-	445	-	-	33	33	34
Marine Environmental Remote Sensing	7	2	2	-	1	1	-		100	
Ocean Surface	79	12	-	4	63	-	-	15	30	55
Ocean Uptake	313	-	-	-	313	-	-	-	100	-
PERD Ice-Ocean Forecasting (Real-Time Ice Forecasting)	200	-	-	40	100	60	-			100
POGO - Partnership for Observations of Global Oceans	21	-	-	-	-	-	21	40	40	20
Scotian Shelf Zooplankton Variability from Optical Plankton ...	19	4	-	-	15	-	-		100	
Section Overhead	80	80	-	-	-	-	-	33	34	33
Shelf Climate Variability	1,165	583	233	117	-	233	-	50	50	
The Feasibility of Stimulating Natural Carbon Dioxide Sinks in the Ocean: A Modelling Study Relevant to Human-Induced Ocean Fertilisation	50	-	-	5	45	-	-		100	
Oceans Science, General - 1210 Total	**5,139**	**1,044**	**255**	**433**	**1,907**	**310**	**1,190**	**22**	**53**	**24**
Phys. & Chem. Oceanogr., Living Res. - 1211										
Fisheries Oceanography	606	15	-	318	212	61	-	50	50	-
Integrated Ecosystem Studies for Modelling Mussel Aquacult.-Environ. Interact.	-	-	-	-	-	-	-	50	50	
ONR-Long-Term Autonomous Measurement of Ocean Dissipation with EPS-MAPPER	16	-	-	2	14	-	-		50	50
Scotia-Emerald Slope Moored Current Measurements	118	24	-	35	59	-	-		50	50
Surface Wave Forecast Model Development	34	3	-	10	10	10	-			100
Ph. & Ch. Oceanogr., Living Res. - 1211 Total	**773**	**55**	**-**	**349**	**283**	**85**	**-**	**39**	**48**	**13**
Biological Oceanography, Ecology - 1212										
A Study of the Short-Term Variability in Primary Production on the Scotian Shelf	40	-	-	-	40	-	-		100	
Carbon Transport and Sequestering in the North Atlantic Ocean	18	-	-	-	18	-	-		100	
Marine Ecosystem Productivity	1,609	805	-	161	644	-	-	-	100	-
Biological Oceanogr., Ecology - 1212 Total	**1,667**	**805**	**-**	**161**	**702**	**-**	**-**	**-**	**100**	**-**
Env. Interactions with Marine Activity - 1213										
Bras d'Or Lakes Studies	17	5	-	5	5	2	-	50	50	
Brooke Ocean Technology	115	58	-	-	-	58	-	50	50	
Coastal Habitat Studies	500	140	10	70	250	30	-	50	50	-
Operational Oceanography	500	33	-	50	234	183	-	-	33	67
Seahorse Ocean Profiler Development	20	8	-	2	2	8	-	33	34	33
Env. Interact's with Marine Activity - 1213 Total	**1,152**	**273**	**9**	**149**	**442**	**279**	**-**	**28**	**43**	**29**
Ocean Climate - 1214										
Additional Costs-CCGS F.G. Creed-Showex '99	1	-	-	-	1	-	-			100
Arctic Ocean Circulation	192	10	-	19	163	-	-	10	78	12
Arctic Ocean Climate Change	115	12	-	12	92	-	-	50	50	
Central Scotian Slope Moored Current Measurements	55	11	-	17	28	-	-	50	50	
Chevron/Texaco - Flemish Pass Moored Current Measurement 11	30	6	-	9	15	-	-		30	70
Climate Change Impact on Energy Sector (Current and Temperature Changes)	312	156	-	62	94	-	-		50	50
Coastal Ocean Climate	466	158	9	121	140	37	-	47	53	-
Encana (Pan Can) - Flemish Pass Moored Current Measurement 111	30	6	-	9	15	-	-		30	70
Iceberg Population	49	5	-	10	34	-	-			100
Ocean Atmosphere Coupling (Impact of Climate Change on Marine Storms and Winds)	196	-	-	20	176	-	-	30	40	30
Ocean Circulation and Variability	672	224	-	45	403	-	-	30	57	13
Ocean Uptake: Physical Pump	75	-	-	8	68	-	-		100	
PetroCanada- Flemish Pass Moored Current Measurement 1	30	6	-	9	15	-	-		30	70
POL Coordination/Management	103	-	-	-	-	-	103	-	42	58
Scotian Shelf Turbulence	12	-	-	1	11	-	-		60	40
Scotian Slope Moored Current Measurements	77	31	-	15	31	-	-		30	70
Validation of Wind Fields (Advanced Wave Models)	142	-	-	14	85	43	-			100
Ocean Climate - 1214 Total	**2,557**	**608**	**8**	**370**	**1,380**	**88**	**103**	**23**	**50**	**27**
Ocean Data, Inf & Advisory Services - 1215										
Ocean Data Information Services	227	-	227	-	-	-	-	33	34	33
Ocean Data, Inf. & Advisory Serv's - 1215 Total	**227**	**-**	**227**	**-**	**-**	**-**	**-**	**33**	**34**	**33**
Oceans Science, Total	**11,515**	**2,785**	**499**	**1,462**	**4,713**	**763**	**1,293**	**21**	**57**	**21**
Aquaculture Science, General - 1220										
Aquaculture Comm & Research Coordination	12	-	-	12	-	-	-	100		
Aquaculture Science Coordination	52	-	-	-	-	-	52	100		
Division Management	194	-	-	-	-	-	194	100		

Aquaculture Science, General - 1220 Total	**258**	**-**	**-**	**12**	**-**	**-**	**246**	**100**	**-**	**-**
Invertebrate Aquaculture - 1222										
Aquaculture Comm & Research Coordination	18	-	-	9	-	-	9	100		
Nutrition - Lipid Research	43	-	-	-	43	-	-	100		
Shellfish Aquaculture and Enhancement	342	-	-	-	342	-	-	100		
Invertebrate Aquaculture - 1222 Total	**403**	**-**	**-**	**9**	**385**	**-**	**9**	**100**	**-**	**-**
Finfish Aquaculture - 1223										
A.C.R.D.P. Administration	59	-	-	-	59	-	-	100		
Aquaculture	214	-	-	-	214	-	-	100		
Aquaculture Biotechnology	179	-	-	-	179	-	-	100		
Aquaculture Comm & Research Coordination	77	-	-	-	77	-	-	100		
Aquatic Animal Health and Oceanography	122	-	-	-	122	-	-	50	50	
Arctic Char Broodstock	150	-	-	-	150	-	-	100		
Atlantic Salmon Aquaculture	67	-	-	-	67	-	-	100		
Crab Meal for Atlantic Salmon	228	-	-	-	228	-	-	100		
Effect of Photoperiod on Atlantic Salmon Maturation	12	-	-	-	12	-	-	100		
European Oyster Broodstock Program, Dartmouth	63	-	-	-	63	-	-	100		
Finfish Reproduction and Broodstock Development	117	-	-	-	117	-	-	100		
General	535	-	-	-	535	-	-	100		
Habitat and Atlantic Salmon Aquaculture	78	-	-	-	78	-	-	100		
Haddock Egg Quality	104	-	-	-	104	-	-	100		
Halibut Egg Quality	94	-	-	-	94	-	-	100		
Halibut Broodstock	138	-	-	-	138	-	-	100		
Marine Finfish Culture	220	-	-	-	220	-	-	100		
Marine Finfish Physiology	61	-	-	-	61	-	-	100		
Nutrition - Lipid Research	129	-	-	-	129	-	-	100		
Finfish Aquaculture - 1223 Total	**2,647**	**-**	**-**	**-**	**2,647**	**-**	**-**	**98**	**2**	**-**
Aquaculture-Environment Interactions - 1225										
Aquaculture Communication & Research Coordination	12	-	-	-	-	-	12	100		
Environmental Biology	91	-	-	9	82	-	-	100		
Environmental Studies for Sustainable Aquaculture	-	-	-	-	-	-	-	100		
Integrated Ecosystem Studies for Modelling Mussel Aquacult.-Environ. Interact.	-	-	-	-	-	-	-	100		
Shellfish Aquaculture and Enhancement	86	-	-	9	77	-	-	100		
Aquaculture-Environment Interactions - 1225 Total	**188**	**-**	**-**	**18**	**159**	**-**	**12**	**100**	**-**	**-**
Aquaculture Science, Total	**3,496**	**-**	**-**	**38**	**3,191**	**-**	**267**	**98**	**2**	**-**
Hydrography, General - 1310										
General	50	-	50	-	-	-	-		10	90
Hydrography, General - 1310 Total	**50**	**-**	**50**	**-**	**-**	**-**	**-**	**-**	**10**	**90**
Hydrographic Surveys - 1311										
Hydrography General	820	820	-	-	-	-	-		10	90
Labrador Survey - PI 2	113	113	-	-	-	-	-		10	90
Hydrographic Surveys - 1311 Total	**932**	**932**	**-**	**-**	**-**	**-**	**-**	**-**	**10**	**90**
Hydrographic Products - 1312										
Aids	25	-	-	-	-	25	-			100
Bridging Staff	164	-	-	-	16	148	-			100
Charts on Datum	172	-	-	-	17	155	-			100
ENC Production	274	-	-	-	-	274	-	-	-	100
Electronic Navigational Charts	-	-	-	-	-	-	-			100
Field Pro Tools	25	-	-	-	13	13	-			100
German Bank Project	125	-	-	-	62	62	-	50	50	-
Hydrogr. Geological Surveys on Georges Bank	257	-	-	-	129	129	-	50	50	
Hydrography, General	4,223	845	211	53	317	1,637	1,161	0	14	85
Labrador Survey - PI 2	38	-	-	-	-	38	-			100
Multibeam Seafloor Mapping	-	-	-	-	-	-	-	10	60	30
Project Administration	260	-	130	-	-	-	130	-	2	98
Revisory	32	-	-	-	-	32	-		20	80
Source Data Base	20	-	20	-	-	-	-		20	80
Terra Remote Sensing Inc.	56	56	-	-	-	-	-			100
Tidal National	35	-	7	17	10	-	-		30	70
Unicorporated Data	15	-	15	-	-	-	-			100
Hydrographic Products - 1312 Total	**5,720**	**1,546**	**770**	**140**	**826**	**2,113**	**325**	**3**	**14**	**82**
Technology Applications & Transfers - 1314										
Data Management Hydrographic Information Network (HIN)	64	-	64	-	-	-	-	33	34	33
Miscellaneous Infrastructure	-	-	-	-	-	-	-			100
Tech. Applications & Transfers - 1314 Total	**64**	**-**	**64**	**-**	**-**	**-**	**-**	**33**	**34**	**33**
Hydrography, Total	**6,767**	**2,479**	**884**	**140**	**826**	**2,113**	**325**	**3**	**14**	**83**
Grand Total	**42,082**	**8,453**	**1,961**	**6,312**	**17,740**	**3,901**	**3,715**	**34**	**47**	**19**

Appendix VI.

Application System Source Code

This appendix contains the source code for the application system developed in this thesis. The system is developed using Visual Basic .NET based on Windows XP. The database is designed in SQL2000. The source codes for Graphic User Interface are eliminated here since the codes are generated by VB.NET Windows Form Designer automatically. There are some VB components that have been pluged in the system for the extra function implementation including Matrix ActiveX Component 3.0 for the matrix calculation, ChartDirector Ver 3.1 for the graphic reports, and finally CST Percent ActiveX V3.6 / VCL V2.1 for the graphic input function.

The main functions of the system include the model construction, data input, analysis, and output report. The code is divided into modules as described in flow chart Figure 4-1. These modules are:

- **Module 1 – Model Construction.** This module implement the first step, model building of proposed procedure that includes three sub-steps 1a, 1b and 2c discussed in Chapter 4.
- **Module 2 – Assessment.** This module implements the second step, alternatives assessment of the proposed procedure in Chapter 4 including sub-steps 2a and 2b.
- **Module 3 – Synthesis.** This module implement step 3a of the third step, decision making of the proposed procedure.
- **Module 4 – Risk Analysis.** This module implements steps 3b and 3c of the third step, decision making of the proposed procedure.

Module 1 Model Construction

This module implements the model building process as the step 1 of proposed procedure described in Chapter 4. The module finish the steps 1a, 1b and 1c shown in the Figure 4-1 and is comprised by six sub-modules including Model Building, Element Input, Institution Definition of Functional Model, Participant Administration, Type of Model Formulation and Participant setup.

Module 1.1 Model Building

This module finishes the model building function of the system. The main functions of the module include creating a new model, editing the contents of hierarchy and setting the participant for the model.

```
Imports System.IO
Public Class DFOmodel
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region
    Dim sConfirm, selResult, selName As String
    Dim selModelID, selNodeID, selParentID, selOption, selType, ModelL1, ModelL2, ModelL3, ModelL4,
currentTab, modelCount As Integer
    Dim selNode, copyNode As TreeNode
    Dim Id(10) As Integer
    Dim selTable As DataTable
    Dim found As Boolean = False
    Private Sub DFOmodel_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
    End Sub
```

'The following codes perform the event raised from the main menu of Module 1.1 including operation on model, editing the elements in hierarchy, and setting the participants for model.

#Region "Menu Item"

'Create a new model

```
    Private Sub MenuNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuNew.Click
        MenuNewModel()
    End Sub
```

'Open a model

```
    Private Sub MenuOpen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuOpen.Click
        MenuOpenModel()
    End Sub
```

'Delete a model

```
    Private Sub MenuDelMod_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDelMod.Click
        If TabModelBuilding.Visible = False Then
            sConfirm = String.Format("No model is chosen!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Else
            sConfirm = String.Format("Are you sure you want to delete this model and its submodel
permenently?")
            selResult = MessageBox.Show(sConfirm, "Confirm Delete !", MessageBoxButtons.OKCancel,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2)
            If selResult = DialogResult.OK Then
                Try
```

```
                Me.UpdateDataSet()
            Catch eUpdate As System.Exception
                System.Windows.Forms.MessageBox.Show(eUpdate.Message)
            End Try
            modelCount = 0
            FindModel(selModelID)
            Dim i As Integer = modelCount
            Do While i >= 0
                DeleteModel(Id(i), objdsModel.TblPairwise)
                DeleteModel(Id(i), objdsModel.TblDatagrid)
                DeleteModel(Id(i), objdsModel.TblOptions)
                DeleteModel(Id(i), objdsModel.TblNodeL2)
                DeleteModel(Id(i), objdsModel.TblNodeL1)
                DeleteModel(Id(i), objdsModel.TblManagerModel)
                DeleteModel(Id(i), objdsModel.TblModel)
                i -= 1
            Loop
            Try
                Me.UpdateDataSet()
            Catch eUpdate As System.Exception
                System.Windows.Forms.MessageBox.Show(eUpdate.Message)
            End Try
            Select Case TabModelBuilding.SelectedIndex
                Case 0
                    ClearEditModel()
                Case 1
                    TreeView2.Nodes.Clear()
                    TreeView3.Nodes.Clear()
                    TreeView4.Nodes.Clear()
                    Me.TreeView2.Nodes.AddRange(New      System.Windows.Forms.TreeNode()      {New
System.Windows.Forms.TreeNode("Goal")})
                    Me.TreeView3.Nodes.AddRange(New      System.Windows.Forms.TreeNode()      {New
System.Windows.Forms.TreeNode("Goal")})
                    Me.TreeView4.Nodes.AddRange(New      System.Windows.Forms.TreeNode()      {New
System.Windows.Forms.TreeNode("Goal")})
                    LisOption2.Items.Clear()
                    LisOption3.Items.Clear()
                    LisOption4.Items.Clear()
                    TabModelBuilding.TabPages(0).Enabled = True
                    TabModelBuilding.TabPages(1).Enabled = False
                    TabModelBuilding.TabPages(2).Enabled = False
                    TabModelBuilding.TabPages(3).Enabled = False
                Case 2
                    TreeView3.Nodes.Clear()
                    TreeView4.Nodes.Clear()
                    Me.TreeView3.Nodes.AddRange(New      System.Windows.Forms.TreeNode()      {New
System.Windows.Forms.TreeNode("Goal")})
                    Me.TreeView4.Nodes.AddRange(New      System.Windows.Forms.TreeNode()      {New
System.Windows.Forms.TreeNode("Goal")})
                    LisOption3.Items.Clear()
                    LisOption4.Items.Clear()
                    TabModelBuilding.TabPages(0).Enabled = True
                    TabModelBuilding.TabPages(1).Enabled = True
                    TabModelBuilding.TabPages(2).Enabled = False
                    TabModelBuilding.TabPages(3).Enabled = False
                Case 3
                    TreeView4.Nodes.Clear()
                    Me.TreeView4.Nodes.AddRange(New      System.Windows.Forms.TreeNode()      {New
System.Windows.Forms.TreeNode("Goal")})
                    LisOption4.Items.Clear()
                    TabModelBuilding.TabPages(0).Enabled = True
                    TabModelBuilding.TabPages(1).Enabled = True
                    TabModelBuilding.TabPages(2).Enabled = True
                    TabModelBuilding.TabPages(3).Enabled = False
            End Select
            TabModelBuilding.SelectedIndex = 0
            currentTab = TabModelBuilding.SelectedIndex
        End If
    End If
End Sub
```

'Close the current editing model

```
    Private Sub MenuClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuClose.Click
        Dim objChanges As DFOdemo.dsModel = New DFOdemo.dsModel()
        objChanges = CType(objdsModel.GetChanges, DFOdemo.dsModel)
```

```
        If (Not (objChanges) Is Nothing) Then
            Dim sConfirm As String = String.Format("Do you want to save the change?")
            selResult  =  MessageBox.Show(sConfirm,  "Comfirm  Save",  MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button3)
            If selResult = DialogResult.Yes Then
                Try
                    Me.UpdateDataSet()
                Catch eUpdate As System.Exception
                    System.Windows.Forms.MessageBox.Show(eUpdate.Message)
                End Try
            End If
            If selResult = DialogResult.No Then
                objdsModel.RejectChanges()
            End If
        End If
        ClearEditModel()
        Me.TabModelBuilding.Visible = False
    End Sub
```

'Save the current editing model

```
    Private Sub MenuSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuSave.Click
        Try
            Me.UpdateDataSet()
        Catch eUpdate As System.Exception
            System.Windows.Forms.MessageBox.Show(eUpdate.Message)
        End Try
    End Sub
```

'Exit the module

```
    Private Sub MenuExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuExit.Click
        Dim objChanges As DFOdemo.dsModel = New DFOdemo.dsModel()
        objChanges = CType(objdsModel.GetChanges, DFOdemo.dsModel)
        If (Not (objChanges) Is Nothing) Then
            Dim sConfirm As String = String.Format("Do you want to save the change?")
            selResult  =  MessageBox.Show(sConfirm,  "Comfirm  Save",  MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button3)
            If selResult = DialogResult.Yes Then
                Try
                    Me.UpdateDataSet()
                Catch eUpdate As System.Exception
                    System.Windows.Forms.MessageBox.Show(eUpdate.Message)
                End Try
                Me.Close()
            End If
            If selResult = DialogResult.No Then
                objdsModel.RejectChanges()
                Me.Close()
            End If
        Else
            Me.Close()
        End If
    End Sub
```

'Insert a node as a child of current node

```
    Private Sub MenuInsertChild_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuInsChild.Click
        MenuInsertChild()
    End Sub
```

'Insert a node as a sibling of current node

```
    Private Sub MenuInsertSibling_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MenuInsSibling.Click
        MenuInsertSibling()
    End Sub
```

'Edit current node

```
    Private Sub MenuEdiNode_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuEdiNode.Click
        MenuEditNode()
    End Sub
```

'Delete current node

```
    Private Sub MenuDeleteNode_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
```

```
MenuDelNode.Click
        sConfirm = String.Format("No node is selected!")
        Select Case TabModelBuilding.SelectedIndex
            Case 0
                If TreeView1.SelectedNode Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelNode(TreeView1.SelectedNode, TreeView1)
                End If
            Case 1
                If TreeView2.SelectedNode Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelNode(TreeView2.SelectedNode, TreeView2)
                End If
            Case 2
                If TreeView3.SelectedNode Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelNode(TreeView3.SelectedNode, TreeView3)
                End If
            Case 3
                If TreeView4.SelectedNode Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelNode(TreeView4.SelectedNode, TreeView4)
                End If
        End Select
    End Sub
```

'Copy current node and its children

```
    Private Sub MenuCopyNode_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuCopNode.Click
        MenuCopyNode()
    End Sub
```

'Paste the nodes copied earlier as the children of current node

```
    Private Sub MenuPasteNode_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuPasNode.Click
        MenuPasteNode()
    End Sub
```

'Add alternative to current model

```
    Private Sub MenuAddAlter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuAddAlt.Click
        MenuAddAlternative()
    End Sub
```

'Delete selected alternative of current model

```
    Private Sub MenuDelAlter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDelAlt.Click
        sConfirm = String.Format("No alternative is selected!")
        Select Case TabModelBuilding.SelectedIndex
            Case 0
                If LisOption1.SelectedItem Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelOption(LisOption1)
                End If
            Case 1
                If LisOption2.SelectedItem Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelOption(LisOption2)
                End If
            Case 2
                If LisOption3.SelectedItem Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
                    DelOption(LisOption3)
                End If
            Case 3
                If LisOption4.SelectedItem Is Nothing Then
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Else
```

```
                DelOption(LisOption4)
            End If
        End Select
    End Sub
```

'Edit information for participants

```
    Private Sub MenuEditManager_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MenuEditManager.Click
        Dim frmManager As New DFOmanager()
        Me.Hide()
        frmManager.ShowDialog()
        Me.Show()
    End Sub
```

'Set up participants for current model

```
    Private Sub MenuSetManager_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuSetManager.Click
        MenuSetupManager()
    End Sub
```

'Edit institution information for functional model

```
    Private Sub MenuIns_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuIns.Click
        Dim frmIns As New DFOmodel_Ins()
        frmIns.ComboIns.Items.Add("Region")
        frmIns.ComboIns.Items.Add("Division")
        frmIns.ComboIns.Items.Add("Program")
        frmIns.ShowDialog(Me)
        Try
            Me.DbConnection.Open()
            Me.daInstitution.Fill(objdsModel)
        Catch fillException As System.Exception
            Throw fillException
        Finally
            Me.DbConnection.Close()
        End Try
    End Sub

    Private Sub MenuAbout_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuAbout.Click
        Dim frmAbout As New DFO_About()
        frmAbout.ShowDialog(Me)
        frmAbout.Dispose()
    End Sub
#End Region
```

'The following code handle the event raised by certain operation other than selecting the menu such as tab switch, double click the element in the hierarchy.

#Region "Event Handle"

'Double click the national level model

```
    Private Sub TreeView1_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TreeView1.DoubleClick
        If Not (TreeView1.SelectedNode) Is Nothing Then
            TreeView1.SelectedNode.Expand()
            EditTree(TreeView1)
        End If
    End Sub
```

'Double click the Regional level model

```
    Private Sub TreeView2_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TreeView2.DoubleClick
        If Not (TreeView2.SelectedNode) Is Nothing Then
            TreeView2.SelectedNode.Expand()
            EditTree(TreeView2)
        End If
    End Sub
```

'Double click the divisional level model

```
    Private Sub TreeView3_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TreeView3.DoubleClick
```

```
        If Not (TreeView3.SelectedNode) Is Nothing Then
            TreeView3.SelectedNode.Expand()
            EditTree(TreeView3)
        End If
    End Sub
```

'Double click the program level model

```
    Private Sub TreeView4_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TreeView4.DoubleClick
        If Not (TreeView4.SelectedNode) Is Nothing Then
            TreeView4.SelectedNode.Expand()
            EditTree(TreeView4)
        End If
    End Sub
```

'Close the current model

```
    Private Sub DFOmodel_Closing(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs)
Handles MyBase.Closing
        Dim objChanges As DFOdemo.dsModel = New DFOdemo.dsModel()
        objChanges = CType(objdsModel.GetChanges, DFOdemo.dsModel)
        If (Not (objChanges) Is Nothing) Then
            Dim sConfirm As String = String.Format("Do you want to save the change?")
            selResult = MessageBox.Show(sConfirm, "Comfirm Save", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
            If selResult = DialogResult.Yes Then
                Try
                    Me.UpdateDataSet()
                Catch eUpdate As System.Exception
                    System.Windows.Forms.MessageBox.Show(eUpdate.Message)
                End Try
            End If
            If selResult = DialogResult.No Then
                objdsModel.RejectChanges()
            End If
        End If
    End Sub
```

'Double click the alternative of the national level model

```
    Private Sub LisOption1_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles LisOption1.DoubleClick
        If Not (LisOption1.SelectedItem) Is Nothing Then
            EditOption(LisOption1)
        End If
    End Sub
```

'Double click the alternative of the regional level model

```
    Private Sub LisOption2_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles LisOption2.DoubleClick
        If Not (LisOption2.SelectedItem) Is Nothing Then
            EditOption(LisOption2)
        End If
    End Sub
```

'Double click the alternative of the divisional level model

```
    Private Sub LisOption3_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles LisOption3.DoubleClick
        If Not (LisOption3.SelectedItem) Is Nothing Then
            EditOption(LisOption3)
        End If
    End Sub
```

'Double click the alternative of the program level model

```
    Private Sub LisOption4_DoubleClick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles LisOption4.DoubleClick
        If Not (LisOption4.SelectedItem) Is Nothing Then
            EditOption(LisOption4)
        End If
    End Sub
```

'Switch among different level models

```
    Private Sub TabModelBuilding_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TabModelBuilding.Click
        sConfirm = Format("You need to choose one Alternative in order to go down to next level model!")
        selResult = Format("You can only view the model one level lower than current model!")
```

```
If selType = 1 Then
    Select Case currentTab
        Case 0
            Select Case TabModelBuilding.SelectedIndex
                Case 1
                    If LisOption1.SelectedItem Is Nothing Then
                        MessageBox.Show(sConfirm,     "Warning     !",     MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                        TabModelBuilding.SelectedIndex = currentTab
                    Else
                        currentTab = TabModelBuilding.SelectedIndex
                        TabModelBuilding.TabPages(0).Enabled = False
                        TabModelBuilding.TabPages(1).Enabled = True
                        MenuUp.Enabled = True
                        EditSubmodel(LisOption1, LisOption2, selModelID, TreeView2)
                        If found Then
                            ModelL2 = selModelID
                        Else
                            TabModelBuilding.SelectedIndex = 0
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(0).Enabled = True
                            TabModelBuilding.TabPages(1).Enabled = False
                            MenuUp.Enabled = False
                            selModelID = ModelL1
                        End If
                    End If
                Case 2
                    MessageBox.Show(selResult,     "Warning     !",     MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                    TabModelBuilding.SelectedIndex = currentTab
                Case 3
                    MessageBox.Show(selResult,     "Warning     !",     MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                    TabModelBuilding.SelectedIndex = currentTab
            End Select
        Case 1
            Select Case TabModelBuilding.SelectedIndex
                Case 0
                    ModelChanged()
                    If found Then
                        currentTab = TabModelBuilding.SelectedIndex
                        TabModelBuilding.TabPages(0).Enabled = True
                        TabModelBuilding.TabPages(1).Enabled = False
                        MenuUp.Enabled = False
                        selModelID = ModelL1
                    Else
                        TabModelBuilding.SelectedIndex = currentTab
                    End If
                Case 2
                    If LisOption2.SelectedItem Is Nothing Then
                        MessageBox.Show(sConfirm,     "Warning     !",     MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                        TabModelBuilding.SelectedIndex = currentTab
                    Else
                        currentTab = TabModelBuilding.SelectedIndex
                        TabModelBuilding.TabPages(1).Enabled = False
                        TabModelBuilding.TabPages(2).Enabled = True
                        EditSubmodel(LisOption2, LisOption3, selModelID, TreeView3)
                        If found Then
                            ModelL3 = selModelID
                        Else
                            TabModelBuilding.SelectedIndex = 1
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(1).Enabled = True
                            TabModelBuilding.TabPages(2).Enabled = False
                            selModelID = ModelL2
                        End If
                    End If
                Case 3
                    MessageBox.Show(selResult,     "Warning     !",     MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                    TabModelBuilding.SelectedIndex = currentTab
            End Select
        Case 2
            Select Case TabModelBuilding.SelectedIndex
                Case 0
                    ModelChanged()
```

```
                        If found Then
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(0).Enabled = True
                            TabModelBuilding.TabPages(1).Enabled = False
                            TabModelBuilding.TabPages(2).Enabled = False
                            TabModelBuilding.TabPages(3).Enabled = False
                            MenuUp.Enabled = False
                            selModelID = ModelL1
                        Else
                            TabModelBuilding.SelectedIndex = currentTab
                        End If
                    Case 1
                        ModelChanged()
                        If found Then
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(0).Enabled = False
                            TabModelBuilding.TabPages(1).Enabled = True
                            TabModelBuilding.TabPages(2).Enabled = False
                            TabModelBuilding.TabPages(3).Enabled = False
                            selModelID = ModelL2
                        Else
                            TabModelBuilding.SelectedIndex = currentTab
                        End If
                    Case 3
                        If LisOption3.SelectedItem Is Nothing Then
                            MessageBox.Show(sConfirm,     "Warning     !",      MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                            TabModelBuilding.SelectedIndex = currentTab
                        Else
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(2).Enabled = False
                            TabModelBuilding.TabPages(3).Enabled = True
                            MenuDown.Enabled = False
                            EditSubmodel(LisOption3, LisOption4, selModelID, TreeView4)
                            If found Then
                                ModelL4 = selModelID
                            Else
                                TabModelBuilding.SelectedIndex = 2
                                currentTab = TabModelBuilding.SelectedIndex
                                TabModelBuilding.TabPages(2).Enabled = True
                                TabModelBuilding.TabPages(3).Enabled = False
                                MenuDown.Enabled = True
                                selModelID = ModelL3
                            End If
                        End If
                End Select
            Case 3
                Select Case TabModelBuilding.SelectedIndex
                    Case 0
                        ModelChanged()
                        If found Then
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(0).Enabled = True
                            TabModelBuilding.TabPages(1).Enabled = False
                            TabModelBuilding.TabPages(2).Enabled = False
                            TabModelBuilding.TabPages(3).Enabled = False
                            MenuUp.Enabled = False
                            selModelID = ModelL1
                        Else
                            TabModelBuilding.SelectedIndex = currentTab
                        End If
                    Case 1
                        ModelChanged()
                        If found Then
                            currentTab = TabModelBuilding.SelectedIndex
                            TabModelBuilding.TabPages(0).Enabled = False
                            TabModelBuilding.TabPages(1).Enabled = True
                            TabModelBuilding.TabPages(2).Enabled = False
                            TabModelBuilding.TabPages(3).Enabled = False
                            selModelID = ModelL2
                        Else
                            TabModelBuilding.SelectedIndex = currentTab
                        End If
                    Case 2
                        ModelChanged()
                        If found Then
                            currentTab = TabModelBuilding.SelectedIndex
```

```
                    TabModelBuilding.TabPages(0).Enabled = False
                    TabModelBuilding.TabPages(1).Enabled = False
                    TabModelBuilding.TabPages(2).Enabled = True
                    TabModelBuilding.TabPages(3).Enabled = False
                    selModelID = ModelL3
                 Else
                    TabModelBuilding.SelectedIndex = currentTab
                 End If
              End Select
           End Select
        End If
        If selType = 2 Then
           Select Case TabModelBuilding.SelectedIndex
              Case 0
                 currentTab = TabModelBuilding.SelectedIndex
                 TabModelBuilding.TabPages(0).Enabled = True
                 TabModelBuilding.TabPages(1).Enabled = False
                 TabModelBuilding.TabPages(2).Enabled = False
                 TabModelBuilding.TabPages(3).Enabled = False
                 MenuUp.Enabled = False
                 selModelID = ModelL1
              Case 1
                 EditSubFuncModel(LisOption2, selModelID, TreeView2)
                 If found Then
                    currentTab = TabModelBuilding.SelectedIndex
                    TabModelBuilding.TabPages(0).Enabled = False
                    TabModelBuilding.TabPages(1).Enabled = True
                    TabModelBuilding.TabPages(2).Enabled = False
                    TabModelBuilding.TabPages(3).Enabled = False
                    ModelL2 = selModelID
                 Else
                    TabModelBuilding.SelectedIndex = currentTab
                 End If
              Case 2
                 EditSubFuncModel(LisOption3, selModelID, TreeView3)
                 If found Then
                    currentTab = TabModelBuilding.SelectedIndex
                    TabModelBuilding.TabPages(0).Enabled = False
                    TabModelBuilding.TabPages(1).Enabled = False
                    TabModelBuilding.TabPages(2).Enabled = True
                    TabModelBuilding.TabPages(3).Enabled = False
                    ModelL3 = selModelID
                 Else
                    TabModelBuilding.SelectedIndex = currentTab
                 End If
              Case 3
                 EditSubFuncModel(LisOption4, selModelID, TreeView4)
                 If found Then
                    currentTab = TabModelBuilding.SelectedIndex
                    TabModelBuilding.TabPages(0).Enabled = False
                    TabModelBuilding.TabPages(1).Enabled = False
                    TabModelBuilding.TabPages(2).Enabled = False
                    TabModelBuilding.TabPages(3).Enabled = True
                    ModelL4 = selModelID
                 Else
                    TabModelBuilding.SelectedIndex = currentTab
                 End If
           End Select
        End If
     End Sub

     Private    Sub    ToolBar_ButtonClick(ByVal    sender    As    System.Object,    ByVal    e    As
System.Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar.ButtonClick
        Select Case ToolBar.Buttons.IndexOf(e.Button)
           Case 0
              MenuNewModel()
           Case 1
              MenuOpenModel()
           Case 2
              Try
                 Me.UpdateDataSet()
              Catch eUpdate As System.Exception
                 System.Windows.Forms.MessageBox.Show(eUpdate.Message)
              End Try
           Case 4
              MenuInsertChild()
           Case 5
```

```
                MenuInsertSibling()
            Case 6
                MenuCopyNode()
            Case 7
                MenuPasteNode()
            Case 9
                MenuAddAlternative()
            Case 10
                Dim frmManager As New DFOmanager()
                Me.Hide()
                frmManager.ShowDialog()
                Me.Show()
            Case 11
                MenuSetupManager()
            Case 13
                Dim frmAbout As New DFO_About()
                frmAbout.ShowDialog(Me)
                frmAbout.Dispose()
        End Select
    End Sub
#End Region
```

'The following codes contain the sub procedures needed in the previous event handle and menu operation. These sub procedures perform the detailed step of most of the functions in this module.

#Region "Sub procedures"

```
    Public Sub MenuNewModel()
        Dim drNewModel As DataRow
        Dim frmNew As New DFOmodel_new()
        frmNew.TextGoal.Visible = True
        frmNew.ShowDialog(Me)
        If frmNew.DialogResult = DialogResult.OK Then
            TabModelBuilding.Visible = True
            If TreeView1.TopNode().Text = "Goal" Then
                ClearEditModel()
                Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text & " - " &
TabModelBuilding.TabPages(0).Text
                Select Case frmNew.modelType.Text
                    Case "Institutional Model"
                        selType = 1
                    Case "Functional Model"
                        selType = 2
                    Case "Integrated Model"
                        selType = 3
                        Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text
                        Me.TabModelBuilding.TabPages(1).Text = " "
                        Me.TabModelBuilding.TabPages(2).Text = " "
                        Me.TabModelBuilding.TabPages(3).Text = " "
                End Select
                NewModel(TreeView1, frmNew.TextGoal.Text, selType, 0)
                ModelL1 = selModelID
            Else
                If TreeView1.TopNode().Text <> frmNew.ComboGoal.Text Then
                    sConfirm = "Do you want to save model " & TreeView1.TopNode.Text & "?"
                    selResult = MessageBox.Show(sConfirm, "Confirm Close", MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
                    If selResult = DialogResult.Yes Then
                        Me.UpdateDataSet()
                        ClearEditModel()
                        Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text & " - " &
TabModelBuilding.TabPages(0).Text
                        Select Case frmNew.modelType.Text
                            Case "Institutional Model"
                                selType = 1
                            Case "Functional Model"
                                selType = 2
                            Case "Integrated Model"
                                selType = 3
                                Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text
                                Me.TabModelBuilding.TabPages(1).Text = " "
                                Me.TabModelBuilding.TabPages(2).Text = " "
                                Me.TabModelBuilding.TabPages(3).Text = " "
                        End Select
                        NewModel(TreeView1, frmNew.TextGoal.Text, selType, 0)
```

```
                ModelL1 = selModelID
            End If
            If selResult = DialogResult.No Then
                objdsModel.RejectChanges()
                ClearEditModel()
                Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text & " - " &
TabModelBuilding.TabPages(0).Text
                Select Case frmNew.modelType.Text
                    Case "Institutional Model"
                        selType = 1
                    Case "Functional Model"
                        selType = 2
                    Case "Integrated Model"
                        selType = 3
                        Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text
                        Me.TabModelBuilding.TabPages(1).Text = " "
                        Me.TabModelBuilding.TabPages(2).Text = " "
                        Me.TabModelBuilding.TabPages(3).Text = " "
                End Select
                NewModel(TreeView1, frmNew.TextGoal.Text, selType, 0)
                ModelL1 = selModelID
            End If
        End If
    End If
  End If
End Sub

Public Sub MenuOpenModel()
    Dim frmNew As New DFOmodel_new()
    frmNew.ComboGoal.Visible = True
    frmNew.ComboGoal.Text = Nothing
    With objdsModel.TblModel
        .DefaultView.RowFilter = "OptionID = 0"
    End With
    frmNew.ComboGoal.DataSource = objdsModel.TblModel.DefaultView
    frmNew.ComboGoal.ValueMember = "ModelType"
    frmNew.ComboGoal.DisplayMember = "ModelName"
    frmNew.ShowDialog(Me)
    If frmNew.DialogResult = DialogResult.OK Then
        TabModelBuilding.Visible = True
        If TreeView1.TopNode().Text = "Goal" Then
            ClearEditModel()
            TreeView1.TopNode().Text = frmNew.ComboGoal.Text
            selModelID = SearchModelId(TreeView1.TopNode, objdsModel.TblModel, 0)
            selType = frmNew.ComboGoal.SelectedValue
            If selType = 3 Then
                Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text
                Me.TabModelBuilding.TabPages(1).Text = " "
                Me.TabModelBuilding.TabPages(2).Text = " "
                Me.TabModelBuilding.TabPages(3).Text = " "
            Else
                Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text & " - " &
TabModelBuilding.TabPages(0).Text
            End If
            ModelL1 = selModelID
            LoadTreeView(selModelID, TreeView1.TopNode)
            LoadOptions(LisOption1)
        Else
            If TreeView1.TopNode().Text <> frmNew.ComboGoal.Text Then
                sConfirm = "Do you want to save model " & TreeView1.TopNode.Text & "?"
                selResult = MessageBox.Show(sConfirm, "Confirm Close", MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
                If selResult = DialogResult.Yes Then
                    Me.UpdateDataSet()
                    ClearEditModel()
                    TreeView1.TopNode().Text = frmNew.ComboGoal.Text
                    selModelID = SearchModelId(TreeView1.TopNode, objdsModel.TblModel, 0)
                    selType = frmNew.ComboGoal.SelectedValue
                    If selType = 3 Then
                        Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text
                        Me.TabModelBuilding.TabPages(1).Text = " "
                        Me.TabModelBuilding.TabPages(2).Text = " "
                        Me.TabModelBuilding.TabPages(3).Text = " "
                    Else
                        Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text & " - " &
TabModelBuilding.TabPages(0).Text
                    End If
```

```
                ModelL1 = selModelID
                LoadTreeView(selModelID, TreeView1.TopNode)
                LoadOptions(LisOption1)
            End If
            If selResult = DialogResult.No Then
                objdsModel.RejectChanges()
                ClearEditModel()
                TreeView1.TopNode().Text = frmNew.ComboGoal.Text
                selType = frmNew.ComboGoal.SelectedValue
                If selType = 3 Then
                    Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text
                    Me.TabModelBuilding.TabPages(1).Text = " "
                    Me.TabModelBuilding.TabPages(2).Text = " "
                    Me.TabModelBuilding.TabPages(3).Text = " "
                Else
                    Me.TabModelBuilding.TabPages(0).Text = frmNew.modelType.Text & " - " &
TabModelBuilding.TabPages(0).Text
                End If
                selModelID = SearchModelId(TreeView1.TopNode, objdsModel.TblModel, 0)
                ModelL1 = selModelID
                LoadTreeView(selModelID, TreeView1.TopNode)
                LoadOptions(LisOption1)
            End If
        End If
    End If
  End If
End Sub

Public Sub MenuInsertChild()
    sConfirm = String.Format("No parent node is selected!")
    Select Case TabModelBuilding.SelectedIndex
        Case 0
            If TreeView1.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddChild(TreeView1)
            End If
        Case 1
            If TreeView2.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddChild(TreeView2)
            End If
        Case 2
            If TreeView3.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddChild(TreeView3)
            End If
        Case 3
            If TreeView4.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddChild(TreeView4)
            End If
    End Select
End Sub

Public Sub MenuInsertSibling()
    sConfirm = String.Format("No node is selected!")
    Select Case TabModelBuilding.SelectedIndex
        Case 0
            If TreeView1.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddSibling(TreeView1)
            End If
        Case 1
            If TreeView2.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddSibling(TreeView2)
            End If
        Case 2
            If TreeView3.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
```

```
                AddSibling(TreeView3)
            End If
        Case 3
            If TreeView4.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                AddSibling(TreeView4)
            End If
    End Select
End Sub

Public Sub MenuEditNode()
    sConfirm = String.Format("No node is selected!")
    Select Case TabModelBuilding.SelectedIndex
        Case 0
            If TreeView1.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                EditTree(TreeView1)
            End If
        Case 1
            If TreeView2.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                EditTree(TreeView2)
            End If
        Case 2
            If TreeView3.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                EditTree(TreeView3)
            End If
        Case 3
            If TreeView4.SelectedNode Is Nothing Then
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                EditTree(TreeView4)
            End If
    End Select
End Sub

Public Sub MenuCopyNode()
    Select Case TabModelBuilding.SelectedIndex
        Case 0
            If Not (TreeView1.SelectedNode Is Nothing) Then
                copyNode = TreeView1.SelectedNode.Clone
            End If
        Case 1
            If Not (TreeView2.SelectedNode Is Nothing) Then
                copyNode = TreeView2.SelectedNode.Clone
            End If
        Case 2
            If Not (TreeView3.SelectedNode Is Nothing) Then
                copyNode = TreeView3.SelectedNode.Clone
            End If
        Case 3
            If Not (TreeView4.SelectedNode Is Nothing) Then
                copyNode = TreeView4.SelectedNode.Clone
            End If
    End Select
End Sub

Public Sub MenuPasteNode()
    Select Case TabModelBuilding.SelectedIndex
        Case 0
            If TreeView1.SelectedNode Is Nothing Then
                sConfirm = String.Format("No parent node is selected!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                TreeView1.SelectedNode.Nodes.Add(copyNode)
                TreeView1.SelectedNode = copyNode
                TreeView1.SelectedNode.Expand()
                PasteNodeToDataset(TreeView1.SelectedNode, TreeView1)
            End If
        Case 1
            If TreeView2.SelectedNode Is Nothing Then
                sConfirm = String.Format("No parent node is selected!")
```

```
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                TreeView2.SelectedNode.Nodes.Add(copyNode)
                TreeView2.SelectedNode = copyNode
                TreeView2.SelectedNode.Expand()
                PasteNodeToDataset(TreeView2.SelectedNode, TreeView2)
            End If
        Case 2
            If TreeView3.SelectedNode Is Nothing Then
                sConfirm = String.Format("No parent node is selected!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                TreeView3.SelectedNode.Nodes.Add(copyNode)
                TreeView3.SelectedNode = copyNode
                TreeView3.SelectedNode.Expand()
                PasteNodeToDataset(TreeView3.SelectedNode, TreeView3)
            End If
        Case 3
            If TreeView4.SelectedNode Is Nothing Then
                sConfirm = String.Format("No parent node is selected!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Else
                TreeView4.SelectedNode.Nodes.Add(copyNode)
                TreeView4.SelectedNode = copyNode
                TreeView4.SelectedNode.Expand()
                PasteNodeToDataset(TreeView4.SelectedNode, TreeView4)
            End If
    End Select
End Sub

Public Sub MenuAddAlternative()
    If TabModelBuilding.Visible Then
        If TreeView1.TopNode.Text <> "Goal" Then
            Dim frmInput As New DFOmodel_Input()
            frmInput.Label1.Visible = True
            frmInput.ShowDialog(Me)
            Dim drAlter As DataRow = objdsModel.TblOptions.NewRow
            If frmInput.DialogResult = DialogResult.OK Then
                Select Case TabModelBuilding.SelectedIndex
                    Case 0
                        LisOption1.Items.Add(frmInput.TexInput.Text)
                    Case 1
                        LisOption2.Items.Add(frmInput.TexInput.Text)
                    Case 2
                        LisOption3.Items.Add(frmInput.TexInput.Text)
                    Case 3
                        LisOption4.Items.Add(frmInput.TexInput.Text)
                End Select
                With drAlter
                    .Item("NodeName") = frmInput.TexInput.Text
                    .Item("ModelID") = selModelID
                    .Item("Priority") = 0
                End With
                objdsModel.TblOptions.Rows.Add(drAlter)
            End If
        End If
    End If
End Sub

Public Sub MenuSetupManager()
    If TabModelBuilding.Visible Then
        If TreeView1.TopNode.Text <> "Goal" Then
            Select Case TabModelBuilding.SelectedIndex
                Case 0
                    selName = TreeView1.TopNode.Text
                Case 1
                    selName = TreeView2.TopNode.Text
                Case 2
                    selName = TreeView3.TopNode.Text
                Case 3
                    selName = TreeView4.TopNode.Text
            End Select
            Dim frmSetup As New DFOmodel_setup()
            frmSetup.LabModel.Text = selName
            frmSetup.LisManager.Items.Clear()
            Dim managerView As DataView = New DataView(objdsModel.TblManager)
            Dim intCount As Integer
```

```
            Dim intCounter As Integer = 0
            managerView.Sort = "Firstname"
            intCount = managerView.Count
            For intCounter = intCounter To intCount - 1
               Dim managerName As String
               managerName    =    managerView.Item(intCounter).Item("Firstname")    &    "    "    &
managerView.Item(intCounter).Item("Lastname")
               frmSetup.LisManager.Items.Add(managerName)
            Next
            Dim strExpr As String = Format("ModelID = " & selModelID)
            Dim foundRows() As DataRow
            foundRows = objdsModel.TblManagerModel.Select(strExpr)
            Dim i As Integer
            frmSetup.LisParticipant.Items.Clear()
            For i = 0 To foundRows.GetUpperBound(0)
               Dim managerName As String
               Dim drManagerName As DataRow
               For Each drManagerName In objdsModel.TblManager
                  If drManagerName.Item("ManagerID") = foundRows(i).Item("ManagerID") Then
                     managerName    =    drManagerName.Item("Firstname")    &    "    "    &
drManagerName.Item("Lastname")
                  End If
               Next
               frmSetup.LisParticipant.Items.Add(managerName)
               frmSetup.LisManager.Items.Remove(managerName)
            Next i
            frmSetup.LisManager.Sorted = True
            frmSetup.LisParticipant.Sorted = True
            frmSetup.ShowDialog(Me)
            If frmSetup.DialogResult = DialogResult.OK Then
               For i = 0 To foundRows.GetUpperBound(0)
                  foundRows(i).Delete()
               Next i
               Try
                  Me.UpdateDataSet()
               Catch eUpdate As System.Exception
                  System.Windows.Forms.MessageBox.Show(eUpdate.Message)
               End Try
               Dim managerID As Integer
               intCounter = 0
               intCount = frmSetup.LisParticipant.Items.Count
               For intCounter = intCounter To intCount - 1
                  Dim drManager As DataRow
                  For Each drManager In objdsModel.TblManager.Rows
                     If  drManager.Item("Firstname")  &  "  "  &  drManager.Item("Lastname")  =
frmSetup.LisParticipant.Items(intCounter) Then
                        managerID = drManager.Item(0)
                     End If
                  Next
                  Dim drNewRelation As DataRow = objdsModel.TblManagerModel.NewRow
                  drNewRelation("ManagerID") = managerID
                  drNewRelation("ModelID") = selModelID
                  objdsModel.TblManagerModel.Rows.Add(drNewRelation)
               Next
               Try
                  Me.UpdateDataSet()
               Catch eUpdate As System.Exception
                  System.Windows.Forms.MessageBox.Show(eUpdate.Message)
               End Try
            End If
         End If
      End If
   End Sub

   Public Sub LoadDataSet()
      Dim objDataSetTemp As DFOdemo.dsModel
      objDataSetTemp = New DFOdemo.dsModel()
      Try
         Me.FillDataSet(objDataSetTemp)
      Catch eFillDataSet As System.Exception
         Throw eFillDataSet
      End Try
      Try
         objdsModel.Clear()
         objdsModel.Merge(objDataSetTemp)
      Catch eLoadMerge As System.Exception
         Throw eLoadMerge
```

```
        End Try
    End Sub

        Public Sub UpdateDataSet()
        Dim objDataSetChanges As DFOdemo.dsModel = New DFOdemo.dsModel()
        objDataSetChanges = CType(objdsModel.GetChanges, DFOdemo.dsModel)
        If (Not (objDataSetChanges) Is Nothing) Then
            Try
                Me.UpdateDataSource(objDataSetChanges)
                objdsModel.Merge(objDataSetChanges)
                objdsModel.AcceptChanges()
            Catch eUpdate As System.Exception
                Throw eUpdate
            End Try
        End If
    End Sub

    Public Sub UpdateDataSource(ByVal ChangedRows As DFOdemo.dsModel)
        Try
            If (Not (ChangedRows) Is Nothing) Then
                Me.DbConnection.Open()
                daDatagrid.Update(ChangedRows)
                daPairwise.Update(ChangedRows)
                daOptions.Update(ChangedRows)
                daNodeL2.Update(ChangedRows)
                daNodeL1.Update(ChangedRows)
                daManagerModel.Update(ChangedRows)
                daModel.Update(ChangedRows)
            End If
        Catch updateException As System.Exception
            Throw updateException
        Finally
            Me.DbConnection.Close()
        End Try
    End Sub

    Public Sub FillDataSet(ByVal dataSet As DFOdemo.dsModel)
        dataSet.EnforceConstraints = False
        Try
            Me.DbConnection.Open()
            Me.daManager.Fill(dataSet)
            Me.daManagerModel.Fill(dataSet)
            Me.daModel.Fill(dataSet)
            Me.daNodeL1.Fill(dataSet)
            Me.daNodeL2.Fill(dataSet)
            Me.daOptions.Fill(dataSet)
            Me.daPairwise.Fill(dataSet)
            Me.daDatagrid.Fill(dataSet)
            Me.daInstitution.Fill(dataSet)
        Catch fillException As System.Exception
            Throw fillException
        Finally
            dataSet.EnforceConstraints = True
            Me.DbConnection.Close()
        End Try
    End Sub

    Public Sub LoadTreeView(ByVal selModelID As Integer, ByVal TopNode As TreeNode)
        Dim i, n As Integer
        Dim all As Boolean = True
        Dim parentrow As DataRow
        Dim ParentTable As DataTable
        ParentTable = objdsModel.Tables("TblNodeL1")
        For Each parentrow In ParentTable.Rows
            If parentrow.Item(2) = selModelID Then
                Dim parentnode As TreeNode
                parentnode = New TreeNode(parentrow.Item(1))
                If parentrow.Item("Priority") > 0 Then
                    parentnode.ImageIndex = 12
                Else
                    all = False
                End If
                TopNode.Nodes.Add(parentnode)
                Dim childrow As DataRow
                Dim childnode As TreeNode
                childnode = New TreeNode()
                For Each childrow In parentrow.GetChildRows("L1ToL2")
```

```
                childnode = parentnode.Nodes.Add(childrow.Item(1))
                If childrow.Item("Priority") > 0 Then
                    childnode.ImageIndex = 12
                Else
                    all = False
                End If
            Next childrow
        End If
    Next parentrow
    If all Then
        TopNode.ImageIndex = 12
    End If
    TopNode.ExpandAll()
End Sub

Public Sub LoadOptions(ByVal list As ListBox)
    list.Items.Clear()
    Dim OptionsView As DataView = New DataView(objdsModel.TblOptions)
    Dim intCount As Integer
    Dim intCounter As Integer = 0
    Dim strExp As String = Format("ModelID = " & selModelID)
    OptionsView.RowFilter = strExp
    intCount = OptionsView.Count
    For intCounter = intCounter To intCount - 1
        list.Items.Add(OptionsView.Item(intCounter).Item(1))
    Next
End Sub

Public Function SearchId(ByVal sNode As TreeNode, ByVal sParentID As Integer, ByVal sTable As DataTable)
As Integer
    Dim drTable As DataRow
    For Each drTable In sTable.Rows
        If drTable.Item(1) = sNode.Text And drTable.Item(2) = sParentID Then
            SearchId = drTable.Item(0)
        End If
    Next
End Function

Public Function SearchModelId(ByVal sNode As TreeNode, ByVal sTable As DataTable, ByVal sOption As
Integer) As Integer
    Dim drTable As DataRow
    For Each drTable In sTable.Rows
        If drTable.Item(1) = sNode.Text And drTable.Item(3) = sOption Then
            SearchModelId = drTable.Item(0)
        End If
    Next
End Function

Private Function SearchTable(ByVal sNode As TreeNode, ByVal tree As TreeView) As DataTable
    If sNode Is tree.TopNode Then
        SearchTable = objdsModel.TblModel
    Else
        If sNode.Parent Is tree.TopNode Then
            SearchTable = objdsModel.TblNodeL1
        Else
            If sNode.Parent.Parent Is tree.TopNode Then
                SearchTable = objdsModel.TblNodeL2
            End If
        End If
    End If
End Function

Private Sub FindModel(ByRef sModelID As Integer)
    Id(modelCount) = sModelID
    Dim dtOption As DataTable = objdsModel.TblOptions
    Dim dtModel As DataTable = objdsModel.TblModel
    Dim foundOptions() As DataRow
    Dim foundModel() As DataRow
    Dim strExpr As String = Format("ModelID = " & sModelID)
    foundOptions = dtOption.Select(strExpr)
    Dim i, j As Integer
    For i = 0 To foundOptions.GetUpperBound(0)
        strExpr = Format("OptionID = " & foundOptions(i).Item(0))
        foundModel = dtModel.Select(strExpr)
        For j = 0 To foundModel.GetUpperBound(0)
            modelCount += 1
            sModelID = foundModel(j).Item(0)
```

```
            FindModel(sModelID)
         Next j
      Next i
   End Sub

   Private Sub DeleteModel(ByVal sModelID As Integer, ByVal sTable As DataTable)
      Dim strExpr As String
      strExpr = Format("ModelID = " & sModelID)
      Dim foundRows() As DataRow
      foundRows = sTable.Select(strExpr)
      Dim i As Integer
      For i = 0 To foundRows.GetUpperBound(0)
         foundRows(i).Delete()
      Next i
   End Sub

   Private Sub PasteNodeToDataset(ByVal sNode As TreeNode, ByVal sTree As TreeView)
      AddNodeToDataset(sNode, sTree)
      Dim aNode As TreeNode
      For Each aNode In sNode.Nodes
         PasteNodeToDataset(aNode, sTree)
      Next
   End Sub

   Private Sub ClearEditModel()
      Dim i As Integer
      For i = 2 To 13
         ToolBar.Buttons(i).Visible = True
      Next
      MenuModel.Visible = True
      MenuNode.Visible = True
      MenuAlternative.Visible = True
      MenuDelMod.Visible = True
      Me.MenuSave.Visible = True
      Me.MenuClose.Visible = True
      TreeView1.Nodes.Clear()
      TreeView2.Nodes.Clear()
      TreeView3.Nodes.Clear()
      TreeView4.Nodes.Clear()
      Me.TreeView1.Nodes.AddRange(New           System.Windows.Forms.TreeNode()            {New
System.Windows.Forms.TreeNode("Goal")})
      Me.TreeView2.Nodes.AddRange(New           System.Windows.Forms.TreeNode()            {New
System.Windows.Forms.TreeNode("Goal")})
      Me.TreeView3.Nodes.AddRange(New           System.Windows.Forms.TreeNode()            {New
System.Windows.Forms.TreeNode("Goal")})
      Me.TreeView4.Nodes.AddRange(New           System.Windows.Forms.TreeNode()            {New
System.Windows.Forms.TreeNode("Goal")})
      LisOption1.Items.Clear()
      LisOption2.Items.Clear()
      LisOption3.Items.Clear()
      LisOption4.Items.Clear()
      TabModelBuilding.TabPages(0).Text = "National"
      TabModelBuilding.TabPages(1).Text = "Regional"
      TabModelBuilding.TabPages(2).Text = "Division"
      TabModelBuilding.TabPages(3).Text = "Program"
      TabModelBuilding.SelectedIndex = 0
      currentTab = TabModelBuilding.SelectedIndex
      TabModelBuilding.TabPages(0).Enabled = True
      TabModelBuilding.TabPages(1).Enabled = False
      TabModelBuilding.TabPages(2).Enabled = False
      TabModelBuilding.TabPages(3).Enabled = False
      MenuUp.Enabled = False
      MenuDown.Enabled = True
   End Sub

   Private Sub EditTree(ByVal sTree As TreeView)
      selTable = SearchTable(sTree.SelectedNode, sTree)
      If sTree.SelectedNode Is sTree.TopNode Then
         selTable = objdsModel.TblModel
         selNodeID = selModelID
      Else
         If sTree.SelectedNode.Parent Is sTree.TopNode Then
            selTable = objdsModel.TblNodeL1
            selNodeID = SearchId(sTree.SelectedNode, selModelID, selTable)
         Else
            selTable = objdsModel.TblNodeL2
            selParentID = SearchId(sTree.SelectedNode.Parent, selModelID, objdsModel.TblNodeL1)
```

```
            selNodeID = SearchId(sTree.SelectedNode, selParentID, selTable)
        End If
    End If
    EditNode(sTree.SelectedNode)
    sTree.SelectedNode.Expand()
    Dim selectDR As DataRow
    For Each selectDR In selTable.Rows
        If selectDR.Item(0) = selNodeID Then
            selectDR.Item(1) = sTree.SelectedNode.Text
        End If
    Next
End Sub

Private Sub EditNode(ByVal sNode As TreeNode)
    Dim frmInput As New DFOmodel_Input()
    frmInput.Label2.Visible = True
    frmInput.TexInput.Text = sNode.Text
    frmInput.ShowDialog(Me)
    If frmInput.DialogResult = DialogResult.OK Then
        sNode.Text = frmInput.TexInput.Text
    End If
    sNode.Expand()
End Sub

Private Sub AddChild(ByVal sTree As TreeView)
    Dim newNode As New TreeNode()
    If sTree.SelectedNode Is sTree.TopNode Then
        EditNode(newNode)
        If newNode.Text <> "" Then
            sTree.SelectedNode.Nodes.Add(newNode)
            sTree.SelectedNode.Expand()
            sTree.SelectedNode = newNode
            AddNodeToDataset(sTree.SelectedNode, sTree)
        End If
    Else
        If sTree.SelectedNode.Parent Is sTree.TopNode Then
            EditNode(newNode)
            If newNode.Text <> "" Then
                sTree.SelectedNode.Nodes.Add(newNode)
                sTree.SelectedNode.Expand()
                sTree.SelectedNode = newNode
                AddNodeToDataset(sTree.SelectedNode, sTree)
            End If
        Else
            sConfirm = String.Format("The selected node is the lowest level of hierarchy already!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End If
End Sub

Private Sub AddSibling(ByVal sTree As TreeView)
    If sTree.SelectedNode Is sTree.TopNode Then
        sConfirm = String.Format("Can't create sibling for the goal of model!")
        MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    Else
        sTree.SelectedNode = sTree.SelectedNode.Parent
        Dim newNode As New TreeNode()
        EditNode(newNode)
        If newNode.Text <> "" Then
            sTree.SelectedNode.Nodes.Add(newNode)
            sTree.SelectedNode.Expand()
            sTree.SelectedNode = newNode
            AddNodeToDataset(sTree.SelectedNode, sTree)
        End If
    End If
End Sub

Private Sub AddNodeToDataset(ByVal sNode As TreeNode, ByVal sTree As TreeView)
    Dim sNodeParentId As Integer
    Dim dtNode As DataTable
    If sNode.Parent Is sTree.TopNode Then
        sNodeParentId = selModelID
        dtNode = objdsModel.TblNodeL1
    Else
        sNodeParentId = SearchId(sNode.Parent, selModelID, objdsModel.TblNodeL1)
        dtNode = objdsModel.TblNodeL2
    End If
```

```
        Dim drNode As DataRow = dtNode.NewRow
        drNode.Item(1) = sNode.Text
        drNode.Item(2) = sNodeParentId
        drNode.Item(3) = selModelID
        drNode.Item(4) = 0
        dtNode.Rows.Add(drNode)
    End Sub

    Private Sub DelNode(ByVal sNode As TreeNode, ByVal sTree As TreeView)
        If sNode Is sTree.TopNode Then
            sConfirm = String.Format("Can't delete the top node of the model")
            selResult    =    MessageBox.Show(sConfirm,    "Warning    !",    MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        Else
            sConfirm = String.Format("The operation can't be reversed, continute?")
            selResult  =  MessageBox.Show(sConfirm,  "Confirm  Delete  !",  MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
            If selResult = DialogResult.Yes Then
                DelNodefromDataset(sNode, sTree)
                sNode.Remove()
                sTree.ExpandAll()
                Try
                    Me.UpdateDataSet()
                Catch eUpdate As System.Exception
                    System.Windows.Forms.MessageBox.Show(eUpdate.Message)
                End Try
            End If
        End If
    End Sub

    Private Sub DelNodefromDataset(ByVal sNode As TreeNode, ByVal sTree As TreeView)
        Dim aNode As TreeNode
        For Each aNode In sNode.Nodes
            DelNodefromDataset(aNode, sTree)
        Next

        Dim nParentId As Integer = selModelID
        Dim dtNode As DataTable = objdsModel.TblNodeL1
        If Not (sNode.Parent Is sTree.TopNode) Then
            nParentId = SearchId(sNode.Parent, selModelID, dtNode)
            dtNode = objdsModel.TblNodeL2
        End If
        Dim ID As Integer = SearchId(sNode, nParentId, dtNode)
        Dim strExpr As String = Format("NodeID = " & ID)
        Dim foundRows() As DataRow
        foundRows = dtNode.Select(strExpr)
        Dim i As Integer
        For i = 0 To foundRows.GetUpperBound(0)
            foundRows(i).Delete()
        Next i
    End Sub

    Private Sub DelOption(ByVal sListBox As ListBox)
        Dim strExpr As String = Format("NodeName = '" & sListBox.SelectedItem & "' And ModelID = " &
selModelID)
        Dim foundRows() As DataRow
        foundRows = objdsModel.TblOptions.Select(strExpr)
        Dim i As Integer
        For i = 0 To foundRows.GetUpperBound(0)
            foundRows(i).Delete()
        Next i
        sListBox.Items.Remove(sListBox.SelectedItem)
    End Sub

    Private Sub EditSubmodel(ByVal list As ListBox, ByVal listNext As ListBox, ByVal sModelID As Integer,
ByVal sTree As TreeView)
        ModelChanged()
        If found Then
            Dim drOption As DataRow
            Dim drModel As DataRow
            Dim modelName As String
            found = False
            For Each drOption In objdsModel.TblOptions
                If drOption.Item("NodeName") = list.SelectedItem And drOption.Item("ModelID") = sModelID
Then
                    selOption = drOption.Item(0)
                End If
```

```
            Next
            For Each drModel In objdsModel.TblModel
                If drModel.Item("OptionID") = selOption Then
                    selModelID = drModel.Item("ModelID")
                    modelName = drModel.Item("ModelName")
                    found = True
                End If
            Next
            If found Then
                sTree.Nodes.Clear()
                sTree.Nodes.AddRange(New                System.Windows.Forms.TreeNode()                 {New
System.Windows.Forms.TreeNode("Goal")})
                listNext.Items.Clear()
                sTree.TopNode().Text = modelName
                LoadTreeView(selModelID, sTree.TopNode)
                LoadOptions(listNext)
            Else
                sTree.Nodes.Clear()
                sTree.Nodes.AddRange(New                System.Windows.Forms.TreeNode()                 {New
System.Windows.Forms.TreeNode("Goal")})
                listNext.Items.Clear()
                sConfirm = String.Format("There is no sublevel model under this alternative, do you want
to create a new one?")
                selResult   =   MessageBox.Show(sConfirm,    "Confirm   !",   MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
                If selResult = DialogResult.Yes Then
                    found = True
                    Dim frmNew As New DFOmodel_new()
                    frmNew.TextGoal.Visible = True
                    frmNew.modelType.Visible = False
                    frmNew.GroupType.Enabled = False
                    frmNew.ShowDialog(Me)
                    If frmNew.DialogResult = DialogResult.OK Then
                        NewModel(sTree, frmNew.TextGoal.Text, selType, selOption)
                    Else
                        found = False
                    End If
                End If
            End If
        End If
    End Sub

    Private Sub EditSubFuncModel(ByVal listNext As ListBox, ByVal sModelID As Integer, ByVal sTree As
TreeView)
        ModelChanged()
        If found Then
            Dim frmNew As New DFOmodel_new()
            frmNew.GroupGoal.Text = "Selected Institution"
            frmNew.ComboGoal.Visible = True
            frmNew.ComboGoal.Text = Nothing ' Clear combo box
            Dim tempModel As DataView = New DataView(objdsModel.TblInstitution)
            Dim strExpr As String = Format("InsLevel = " & TabModelBuilding.SelectedIndex)
            tempModel.RowFilter = strExpr
            frmNew.ComboGoal.DataSource = tempModel
            frmNew.ComboGoal.DisplayMember = "InsName"
            frmNew.ComboGoal.ValueMember = "InsID"
            frmNew.ShowDialog(Me)
            If frmNew.DialogResult = DialogResult.OK And frmNew.ComboGoal.SelectedValue > 0 Then
                selOption = frmNew.ComboGoal.SelectedValue
                Dim drOption As DataRow
                Dim drModel As DataRow
                Dim modelName As String
                found = False
                For Each drModel In objdsModel.TblModel
                    If drModel.Item("OptionID") = selOption Then
                        selModelID = drModel.Item("ModelID")
                        modelName = drModel.Item("ModelName")
                        found = True
                    End If
                Next
                If found Then
                    sTree.Nodes.Clear()
                    sTree.Nodes.AddRange(New            System.Windows.Forms.TreeNode()              {New
System.Windows.Forms.TreeNode("Goal")})
                    listNext.Items.Clear()
                    sTree.TopNode().Text = modelName
                    LoadTreeView(selModelID, sTree.TopNode)
```

```
                LoadOptions(listNext)
            Else
                sTree.Nodes.Clear()
                sTree.Nodes.AddRange(New          System.Windows.Forms.TreeNode()           {New
System.Windows.Forms.TreeNode("Goal")})
                listNext.Items.Clear()
                sConfirm = String.Format("There is no sublevel model under this alternative, do you
want to create a new one?")
                selResult  =  MessageBox.Show(sConfirm,  "Confirm  !",  MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
                If selResult = DialogResult.Yes Then
                    found = True
                    Dim frmNew1 As New DFOmodel_new()
                    frmNew1.TextGoal.Visible = True
                    frmNew1.modelType.Visible = False
                    frmNew1.GroupType.Enabled = False
                    frmNew1.ShowDialog(Me)
                    If frmNew1.DialogResult = DialogResult.OK Then
                        NewModel(sTree, frmNew.TextGoal.Text, selType, selOption)
                    Else
                        found = False
                    End If
                End If
            End If
        End If
    End If
End Sub

Private Sub NewModel(ByVal sTree As TreeView, ByVal name As String, ByVal type As Integer, ByVal sOption
As Integer)
    sTree.TopNode().Text = name
    Dim drNewModel As DataRow
    drNewModel = objdsModel.TblModel.NewRow()
    drNewModel("ModelName") = name
    drNewModel("ModelType") = type
    drNewModel("OptionID") = sOption
    drNewModel("Allocation") = 0
    drNewModel("SchemeID") = 0
    objdsModel.TblModel.Rows.Add(drNewModel)
    Dim counter As Integer = objdsModel.TblModel.Rows.Count - 1
    selModelID = objdsModel.TblModel.Rows(counter).Item(0)
End Sub

Private Sub EditOption(ByVal sOption As ListBox)
    Dim frmInput As New DFOmodel_Input()
    frmInput.Label1.Visible = True
    frmInput.TexInput.Text = sOption.SelectedItem
    frmInput.ShowDialog(Me)
    Dim drAlter As DataRow
    If frmInput.DialogResult = DialogResult.OK Then
        For Each drAlter In objdsModel.TblOptions.Rows
            If  drAlter.Item("NodeName")  =  sOption.SelectedItem  And  drAlter.Item("ModelID")  =
selModelID Then
                drAlter.Item("NodeName") = frmInput.TexInput.Text
            End If
        Next
        LoadOptions(sOption)
    End If
End Sub

Private Sub ModelChanged()
    Dim objDataSetChanges As DFOdemo.dsModel = New DFOdemo.dsModel()
    objDataSetChanges = CType(objdsModel.GetChanges, DFOdemo.dsModel)
    found = True
    If (Not (objDataSetChanges) Is Nothing) Then
        sConfirm = Format("You need to save the change made to current model before going to other
model, continute?")
        selResult = MessageBox.Show(sConfirm, "Change Model Confirm ", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
        If selResult = DialogResult.OK Then
            Try
                Me.UpdateDataSource(objDataSetChanges)
                objdsModel.Merge(objDataSetChanges)
                objdsModel.AcceptChanges()
            Catch eUpdate As System.Exception
                Throw eUpdate
            End Try
```

```
            Else
                found = False
            End If
        End If
    End Sub
#End Region
```

Module 1.2 Element Input

This module is a sub-module designed for editing the element of hierarchy.

```
Public Class DFOmodel_Input
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Private Sub ButSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ButSave.Click
        Me.DialogResult = DialogResult.OK
        Me.Close()
    End Sub

    Private Sub ButCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ButCancel.Click
        Me.DialogResult = DialogResult.Cancel
        Me.Close()
    End Sub
End Class
```

Module 1.3 Institution Definition of Functional Model

This module is designed to setup the institution for functional model formulation.

```
Public Class DFOmodel_Ins
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim selLevel As Integer
    Private Sub DFOmodel_Ins_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        ComboIns.SelectedIndex = 0
        ResetDatagrid()
        SetGirdIns()
    End Sub

    Private Sub ResetDatagrid()
        With GridIns
            .BackgroundColor = SystemColors.InactiveCaptionText
            .CaptionText = ""
            .CaptionBackColor = SystemColors.ActiveCaption
            .TableStyles.Clear()
            .ResetAlternatingBackColor()
            .ResetBackColor()
            .ResetForeColor()
            .ResetGridLineColor()
            .ResetHeaderBackColor()
            .ResetHeaderFont()
            .ResetHeaderForeColor()
            .ResetSelectionBackColor()
            .ResetSelectionForeColor()
            .ResetText()
            .BorderStyle = BorderStyle.Fixed3D
            .DataSource = Nothing
```

```
        End With
    End Sub

    Private Sub SetGirdIns()
        If ComboIns.SelectedItem = "Region" Then
            selLevel = 1
            LblLevel.Text = "Level 1:"
        End If
        If ComboIns.SelectedItem = "Division" Then
            selLevel = 2
            LblLevel.Text = "Level 2:"
        End If
        If ComboIns.SelectedItem = "Program" Then
            selLevel = 3
            LblLevel.Text = "Level 3:"
        End If
        With objdsAssessment.TblInstitution
            .DefaultView.RowFilter = "InsLevel =" & selLevel
        End With
        With GridIns
            .DataSource = objdsAssessment.TblInstitution.DefaultView
            .BackColor = Color.GhostWhite
            .BackgroundColor = Color.GhostWhite
            .CaptionBackColor = Color.RoyalBlue
            .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
            .CaptionText = ComboIns.SelectedItem
            .Font = New Font("Tahoma", 12.0!)
            .ReadOnly = False
        End With
        Dim TableStyle1 As New DataGridTableStyle()
        With TableStyle1
            .GridLineColor = Color.RoyalBlue
            .MappingName = "TblInstitution"
            .PreferredColumnWidth = 150
            .PreferredRowHeight = 20
        End With
        Dim ColumStyle As New DataGridTextBoxColumn()
        With ColumStyle
            .HeaderText = "Institution"
            .MappingName = "InsName"
            .Alignment = HorizontalAlignment.Center
            .Width = 250
        End With
        Dim ColumStyle0 As New DataGridTextBoxColumn()
        With ColumStyle0
            .HeaderText = "Level"
            .MappingName = "InsLevel"
            .Alignment = HorizontalAlignment.Center
        End With
        TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle, ColumStyle0})
        GridIns.TableStyles.Add(TableStyle1)
    End Sub

    Public Sub LoadDataSet()
        Dim objDataSetTemp As DFOdemo.dsAssessment
        objDataSetTemp = New DFOdemo.dsAssessment()
        Try
            Me.FillDataSet(objDataSetTemp)
        Catch eFillDataSet As System.Exception
            Throw eFillDataSet
        End Try
        Try
            objdsAssessment.Clear()
            objdsAssessment.Merge(objDataSetTemp)
        Catch eLoadMerge As System.Exception
            Throw eLoadMerge
        End Try
    End Sub

    Public Sub UpdateDataSet()
        Dim objDataSetChanges As DFOdemo.dsAssessment = New DFOdemo.dsAssessment()
        objDataSetChanges = CType(objdsAssessment.GetChanges, DFOdemo.dsAssessment)
        If (Not (objDataSetChanges) Is Nothing) Then
            Try
                Me.UpdateDataSource(objDataSetChanges)
                objdsAssessment.Merge(objDataSetChanges)
                objdsAssessment.AcceptChanges()
```

```
            Catch eUpdate As System.Exception
                Throw eUpdate
            End Try
        End If
    End Sub

    Public Sub UpdateDataSource(ByVal ChangedRows As DFOdemo.dsAssessment)
        Try
            If (Not (ChangedRows) Is Nothing) Then
                Me.DbConnection.Open()
                daInstitution.Update(ChangedRows)
            End If
        Catch updateException As System.Exception
            Throw updateException
        Finally
            Me.DbConnection.Close()
        End Try
    End Sub

    Public Sub FillDataSet(ByVal dataSet As DFOdemo.dsAssessment)
        dataSet.EnforceConstraints = False
        Try
            Me.DbConnection.Open()
            Me.daInstitution.Fill(dataSet)
        Catch fillException As System.Exception
            Throw fillException
        Finally
            dataSet.EnforceConstraints = True
            Me.DbConnection.Close()
        End Try
    End Sub

    Private Sub ComboIns_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles ComboIns.SelectedIndexChanged
        ResetDatagrid()
        SetGirdIns()
    End Sub

    Private Sub DFOmodel_Ins_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
        Me.UpdateDataSet()
    End Sub
End Class
```

Module 1.4 Participant Administration

The function of this module is to edit the personal information of participants.

```
Public Class DFOmanager
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim searchID As New Integer()
    Private Sub DFOmanager_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        Me.objdsManager_PositionChanged()
    End Sub

    Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnSearch.Click
        Dim drModel As DataRow
        Dim strExpr As String = Format("ManagerID = " & CInt(editManagerID.Text()))
        Dim foundRows() As DataRow
        foundRows = objdsManager.TblManagerModel.Select(strExpr)
        Dim i As Integer
        For i = 0 To foundRows.GetUpperBound(0)
            For Each drModel In objdsManager.TblModel.Rows
```

```
            If drModel.Item("ModelID") = foundRows(i).Item("ModelID") Then
                lstModelName.Items.Add(drModel("ModelName"))
            End If
        Next
    Next i
End Sub

Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnNew.Click
    Try
        Me.BindingContext(objdsManager, "TblManager").EndCurrentEdit()
        Me.BindingContext(objdsManager, "TblManager").AddNew()
    Catch eEndEdit As System.Exception
        System.Windows.Forms.MessageBox.Show(eEndEdit.Message)
    End Try
    Me.objdsManager_PositionChanged()
End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnSave.Click
    Try
        Me.UpdateDataSet()
    Catch eUpdate As System.Exception
        System.Windows.Forms.MessageBox.Show(eUpdate.Message)
    End Try
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
    Dim sConfirm As String = String.Format("Are you sure you want to delete this participant?")
    If MessageBox.Show(sConfirm, "Comfirm Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2) = DialogResult.Yes Then
        If (Me.BindingContext(objdsManager, "TblManager").Count > 0) Then
            searchID = Me.BindingContext(objdsManager, "TblManager").Current("ManagerID")
            Me.BindingContext(objdsManager, "TblManager").RemoveAt(Me.BindingContext(objdsManager,
"TblManager").Position)
            Dim drManagerModel As DataRow
            Me.BindingContext(objdsManager, "TblManagerModel").Position = 0
            For Each drManagerModel In objdsManager.TblManagerModel.Rows
                If drManagerModel.Item("ManagerID") = searchID Then
                    objdsManager.TblManagerModel.RemoveTblManagerModelRow(drManagerModel)
                End If
            Next
            Me.objdsManager_PositionChanged()
        End If
    End If
End Sub

Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCancel.Click
    Me.BindingContext(objdsManager, "TblManager").CancelCurrentEdit()
    Me.objdsManager_PositionChanged()
End Sub

Private Sub btnNavFirst_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnNavFirst.Click
    Me.BindingContext(objdsManager, "TblManager").Position = 0
    Me.objdsManager_PositionChanged()
End Sub

Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnLast.Click
    Me.BindingContext(objdsManager,                    "TblManager").Position                =
(Me.objdsManager.Tables("TblManager").Rows.Count - 1)
    Me.objdsManager_PositionChanged()
End Sub

Private Sub btnNavPrev_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnNavPrev.Click
    Me.BindingContext(objdsManager,  "TblManager").Position  =  (Me.BindingContext(objdsManager,
"TblManager").Position - 1)
    Me.objdsManager_PositionChanged()
End Sub

Private Sub btnNavNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnNavNext.Click
    Me.BindingContext(objdsManager,  "TblManager").Position  =  (Me.BindingContext(objdsManager,
```

```
"TblManager").Position + 1)
        Me.objdsManager_PositionChanged()
    End Sub

    Private Sub objdsManager_PositionChanged()
        lstModelName.Items.Clear()
        Me.lblNavLocation.Text   =   (((Me.BindingContext(objdsManager,   "TblManager").Position   +
1).ToString + " of  ") _
                + Me.BindingContext(objdsManager, "TblManager").Count.ToString)
    End Sub

    Public Sub UpdateDataSet()
        Dim objDataSetChanges As DFOdemo.dsManager = New DFOdemo.dsManager()
        Me.BindingContext(objdsManager, "TblManager").EndCurrentEdit()
        objDataSetChanges = CType(objdsManager.GetChanges, DFOdemo.dsManager)
        If (Not (objDataSetChanges) Is Nothing) Then
            Try
                Me.UpdateDataSource(objDataSetChanges)
                objdsManager.Merge(objDataSetChanges)
                objdsManager.AcceptChanges()
            Catch eUpdate As System.Exception
                Throw eUpdate
            End Try
        End If
    End Sub

    Public Sub LoadDataSet()
        Dim objDataSetTemp As DFOdemo.dsManager
        objDataSetTemp = New DFOdemo.dsManager()
        Try
            Me.FillDataSet(objDataSetTemp)
        Catch eFillDataSet As System.Exception
            Throw eFillDataSet
        End Try
        Try
            objdsManager.Clear()
            objdsManager.Merge(objDataSetTemp)
        Catch eLoadMerge As System.Exception
            Throw eLoadMerge
        End Try
    End Sub

    Public Sub UpdateDataSource(ByVal ChangedRows As DFOdemo.dsManager)
        Try
            If (Not (ChangedRows) Is Nothing) Then
                Me.DbConnection.Open()
                daManager.Update(ChangedRows)
                daManagerModel.Update(ChangedRows)
            End If
        Catch updateException As System.Exception
            Throw updateException
        Finally
            Me.DbConnection.Close()
        End Try
    End Sub

    Public Sub FillDataSet(ByVal dataSet As DFOdemo.dsManager)
        dataSet.EnforceConstraints = False
        Try
            Me.DbConnection.Open()
            Me.daManager.Fill(dataSet)
            Me.daManagerModel.Fill(dataSet)
            Me.daModel.Fill(dataSet)
        Catch fillException As System.Exception
            Throw fillException
        Finally
            dataSet.EnforceConstraints = True
            Me.DbConnection.Close()
        End Try
    End Sub

    Private Sub MenuExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.Close()
    End Sub

    Private Sub MenuAbout_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuAbout.Click
```

```
        Dim frmAbout As New DFO_About()
        frmAbout.ShowDialog(Me)
        frmAbout.Dispose()
    End Sub
End Class
```

Module 1.5 Type of Model Formulation

This module decide the formulation type of model at the beginning of model construction process.

```
Public Class DFOmodel_new
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim sConfirm As String
    Dim modelValue As String
    Dim found As Boolean = False
    Private Sub DFOmodel_new_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Me.DbConnection.Open()
        Me.daModel.Fill(objdsModel)
        Me.DbConnection.Close()
        If TextGoal.Visible = True Then
            GroupType.Visible = True
        End If
    End Sub

    Private Sub ButOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ButOK.Click
        If TextGoal.Visible = True Then
            If TextGoal.Text = "" Then
                sConfirm = String.Format("Model goal needed to be decided before building the model!")
                MessageBox.Show(sConfirm, "Warning Message", MessageBoxButtons.OK,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button1)
            Else
                Dim drModel As DataRow
                For Each drModel In objdsModel.TblModel
                    If drModel.Item("ModelName") = TextGoal.Text Then
                        found = True
                    End If
                Next
                If found Then
                    sConfirm = String.Format("Model " & TextGoal.Text & " already exist!")
                    MessageBox.Show(sConfirm, "Warning Message", MessageBoxButtons.OK,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button1)
                Else
                    If RadButIns.Checked = True Then
                        modelType.Text = RadButIns.Text
                        modelClose()
                    Else
                        If RadButFun.Checked = True Then
                            modelType.Text = RadButFun.Text
                            modelClose()
                        Else
                            If RadButInt.Checked = True Then
                                modelType.Text = RadButInt.Text
                                modelClose()
                            Else
                                sConfirm = String.Format("Model type needed to be decided before building
the model!")
                                MessageBox.Show(sConfirm, "Warning Message", MessageBoxButtons.OK,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button1)
                            End If
                        End If
                    End If
                End If
            End If
        End If
        If ComboGoal.Visible = True Then
            modelValue = ComboGoal.SelectedValue
            Select Case modelValue
                Case "1"
```

```
                modelType.Text = "Institutional Model"
            Case "2"
                modelType.Text = "Functional Model"
            Case "3"
                modelType.Text = "Integrated Model"
        End Select
        modelClose()
    End If
End Sub

Private Sub modelClose()
    Me.Close()
End Sub

Private Sub ButCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ButCancel.Click
    Me.Close()
End Sub
End Class
```

Module 1.6 Participant setup

This module is designed to setup the participant for a specific model.

```
Public Class DFOmodel_setup
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim sConfirm As String
    Private Sub modelClose()
        Me.DialogResult = DialogResult.OK
        Me.Close()
    End Sub

    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCancel.Click
        Me.DialogResult = DialogResult.Cancel
        Me.Close()
    End Sub

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnSave.Click
        modelClose()
    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
        If LisManager.SelectedItem = "" Then
            sConfirm = String.Format("No manager was selected!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Else
            LisParticipant.Items.Add(LisManager.SelectedItem)
            LisManager.Items.Remove(LisManager.SelectedItem)
        End If
    End Sub

    Private Sub btnAddAll_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnAddAll.Click
        Dim intCount As Integer
        Dim intCounter As Integer = 0
        intCount = LisManager.Items.Count
        For intCounter = intCounter To intCount - 1
            LisParticipant.Items.Add(LisManager.Items(intCounter))
        Next
        LisManager.Items.Clear()
    End Sub

    Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
        If LisParticipant.SelectedItem = "" Then
            sConfirm = String.Format("No participant was selected!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
```

```
        Else
            LisManager.Items.Add(LisParticipant.SelectedItem)
            LisParticipant.Items.Remove(LisParticipant.SelectedItem)
        End If
    End Sub

    Private Sub btnDeleteAll_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnDeleteAll.Click
        Dim intCount As Integer
        Dim intCounter As Integer = 0
        intCount = LisParticipant.Items.Count
        For intCounter = intCounter To intCount - 1
            LisManager.Items.Add(LisParticipant.Items(intCounter))
        Next
        LisParticipant.Items.Clear()
    End Sub
End Class
```

Module 2 Assessment

Module 2 implements the second step of proposed procedure discussed in Chapter 4. It includes function of pairwise comparison, datagrid evaluation, and priorities calculation. The module is divided into two sub-modules including Assessment Process and Rating Scale Setup.

Module 2.1 Assessment Process

This module performs the main function of step 2a and 2b in the flow chart shown in Figure 4-1. Functions including pairwise comparison, datagrid evaluation, and priorities calculation can be done by this module.

```
Imports System.Math
Imports BluebitMatrix30
Public Class DFOassess
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim sConfirm, selResult, selName As String
    Dim selModelID, selScheme, selParentID, ChildCount, preRow, preColumn, curRow, curColumn, index,
MaxValue, MinValue As Integer
    Dim selManager As Integer = 0
    Dim ChildOfNode(20) As String
    Dim ShowID(10) As Integer
    Dim ShowValue(10) As Decimal
    Dim selNode As TreeNode
    Dim selTable As DataTable
    Dim found As Boolean = False
    Dim Ready As Boolean = True
    Dim temp As New Matrix()
    Dim NodeData As New Matrix()
    Dim OptionData As New Matrix()
    Private Sub DFOassess_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        Me.Show()
        Dim frmNew As New DFOmodel_new()
        frmNew.ComboGoal.Visible = True
        frmNew.ComboGoal.Text = Nothing
        frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
        frmNew.ComboGoal.DisplayMember = "ModelName"
        frmNew.ComboGoal.ValueMember = "ModelType"
        frmNew.ShowDialog(Me)
        If frmNew.DialogResult = DialogResult.OK Then
            TreeView1.TopNode().Text = frmNew.ComboGoal.Text
            SearchModelId()
```

```
            LoadTreeView(selModelID, TreeView1.TopNode)
            LoadOptions()
        Else
            Me.Close()
        End If
    End Sub
```

'The following code perform the event raised from the main menu of Module 2.1 including operations about data input, scale setting and priorities calculation.

#Region "Menu Item"

```
    Private Sub MenuOpen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuOpen.Click
        MenuOpenModel()
    End Sub

    Private Sub MenuSave_Click (See same Sub in Module 1)

    Private Sub MenuAbout_Click (See same Sub in Module 1)

    Private Sub MenuExit_Click (See same Sub in Module 1)
```

' Delete the pairwise comparison data for children under current node

```
    Private Sub MenuDelPair_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDelPair.Click
        If TreeView1.SelectedNode Is Nothing Then
            sConfirm = Format("You need to choose a node to delete the judgement under it!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Else
            If TreeView1.SelectedNode Is TreeView1.TopNode Then
                selParentID = selModelID
                sConfirm = Format("Are you sure you want to delete all the judgement made with respect to
" & TreeView1.SelectedNode.Text & " ?")
                selResult = MessageBox.Show(sConfirm, "Delete Confirm", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
                If selResult = DialogResult.OK Then
                    Dim strExpr As String
                    strExpr = Format("ParentID = " & selModelID)
                    Dim foundRows() As DataRow
                    foundRows = objdsAssessment.TblPairwise.Select(strExpr)
                    Dim i As Integer
                    For i = 0 To foundRows.GetUpperBound(0)
                        foundRows(i).Delete()
                    Next i
                End If
            Else
                If TreeView1.SelectedNode.Parent Is TreeView1.TopNode Then
                    selParentID = SearchId(TreeView1.SelectedNode, selModelID, objdsAssessment.TblNodeL1)
                    sConfirm = Format("Are you sure you want to delete all the judgement made with respect
to " & TreeView1.SelectedNode.Text & " ?")
                    selResult = MessageBox.Show(sConfirm, "Delete Confirm", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
                    If selResult = DialogResult.OK Then
                        Dim strExpr As String
                        strExpr = Format("ParentID = " & selModelID)
                        Dim foundRows() As DataRow
                        foundRows = objdsAssessment.TblPairwise.Select(strExpr)
                        Dim i As Integer
                        For i = 0 To foundRows.GetUpperBound(0)
                            foundRows(i).Delete()
                        Next i
                    End If
                Else
                    sConfirm = Format("There is no judgement under this node!")
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                End If
            End If
        End If
    End Sub
```

' Delete the pairwise comparison data for all node

```
    Private Sub MenuDelPairAll_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDelPairAll.Click
```

```
        sConfirm = Format("Are you sure you want to delete all the judgement made for " &
TreeView1.TopNode.Text & " ?")
        selResult = MessageBox.Show(sConfirm, "Delete Confirm", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
        If selResult = DialogResult.OK Then
            Dim strExpr As String
            strExpr = Format("ModelID = " & selModelID)
            Dim foundRows() As DataRow
            foundRows = objdsAssessment.TblPairwise.Select(strExpr)
            Dim i As Integer
            For i = 0 To foundRows.GetUpperBound(0)
                foundRows(i).Delete()
            Next i
        End If
    End Sub
```

' Delete the datagrid data for current alternative

```
    Private Sub MenuDelData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDelData.Click
        If LisOptions.SelectedItem Is Nothing Then
            sConfirm = Format("You need to choose a alternative to delete its judgement!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Else
            sConfirm = Format("Are you sure you want to delete the judgement made for " &
LisOptions.SelectedItem & " ?")
            selResult = MessageBox.Show(sConfirm, "Delete Confirm", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
            If selResult = DialogResult.OK Then
                Dim strExpr As String
                strExpr = Format("ModelID = " & selModelID & "And NodeName = " & LisOptions.SelectedItem)
                Dim foundRows() As DataRow
                foundRows = objdsAssessment.TblPairwise.Select(strExpr)
                Dim i, j As Integer
                For i = 0 To foundRows.GetUpperBound(0)
                    For j = 5 To 24
                        foundRows(i).Item(j) = 0
                    Next j
                Next i
            End If
        End If
    End Sub
```

' Delete the datagrid data for all alternative

```
    Private Sub MenuDelDataAll_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDelDataAll.Click
        sConfirm = Format("Are you sure you want to delete the judgement made for all the options of "
& TreeView1.SelectedNode.Text & " ?")
        selResult = MessageBox.Show(sConfirm, "Delete Confirm", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
        If selResult = DialogResult.OK Then
            Dim strExpr As String
            strExpr = Format("ModelID = " & selModelID)
            Dim foundRows() As DataRow
            foundRows = objdsAssessment.TblPairwise.Select(strExpr)
            Dim i As Integer
            For i = 0 To foundRows.GetUpperBound(0)
                foundRows(i).Delete()
            Next i
        End If
    End Sub
```

' Calculate local priorities

```
    Private Sub MenuCalPair_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuCalPair.Click
        Ready = True
        CalPairwise(TreeView1.TopNode, selModelID)
        If Ready Then
            sConfirm = Format("The calculation complete sucessfully!")
            MessageBox.Show(sConfirm, "Confirm !", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
            sConfirm = Format("There are not enough information to finish the calculation!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub
```

' Calculate global priorities

```
    Private Sub MenuCalData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuCalData.Click
        MenuCalDatagrid()
    End Sub
#End Region
```

'The following codes contain the sub procedures needed in the previous event handle and menu operation. These sub procedures perform the detailed step of most of the functions in this module.

#Region "Sub Procedure"

```
    Public Sub MenuOpenModel()
        Dim frmNew As New DFOmodel_new()
        frmNew.ComboGoal.Visible = True
        frmNew.ComboGoal.Text = Nothing
        frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
        frmNew.ComboGoal.DisplayMember = "ModelName"
        frmNew.ComboGoal.ValueMember = "ModelType"
        frmNew.ShowDialog(Me)
        If frmNew.DialogResult = DialogResult.OK Then
            If TreeView1.TopNode().Text = "Goal" Then
                TreeView1.TopNode().Text = frmNew.ComboGoal.Text
                SearchModelId()
                LoadTreeView(selModelID, TreeView1.TopNode)
                LoadOptions()
            Else
                If TreeView1.TopNode().Text <> frmNew.ComboGoal.Text Then
                    CheckChange()
                    ClearEditModel()
                    TreeView1.TopNode().Text = frmNew.ComboGoal.Text
                    SearchModelId()
                    LoadTreeView(selModelID, TreeView1.TopNode)
                    LoadOptions()
                End If
            End If
        End If
    End Sub

    Public Sub MenuCalDatagrid()
        Ready = True
        Dim strExpr As String
        strExpr = Format("ModelID = " & selModelID)
        Dim foundRows() As DataRow
        foundRows = objdsAssessment.TblNodeL1.Select(strExpr)
        Dim selRow As Integer = foundRows.GetUpperBound(0)
        Dim tempTotal As New Matrix()
        temp.Size(selRow + 1, 6)
        NodeData.Size(selRow + 1, 6)
        OptionData.Size(selRow + 1, 6)
        tempTotal.Size(selRow + 1, 6)
        Dim temp0 As New Matrix()
        temp0.Size(1, selRow + 1)
        foundRows = objdsAssessment.TblDatagrid.Select(strExpr)
        index = foundRows.GetUpperBound(0)
        Dim priority(index + 1) As Double
        Dim selOptions(index + 1) As Integer
        Dim i, j As Integer
        For i = 0 To index
            selOptions(i) = foundRows(i).Item(0)
            curRow = i
            curColumn = 1
            OptionData.Clear()
            PriorityL1(TreeView1.TopNode.FirstNode, 0)
            tempTotal = tempTotal.Plus(OptionData)
        Next
        For i = 0 To index
            selOptions(i) = foundRows(i).Item(0)
            curRow = i
            curColumn = 1
            temp.Clear()
            NodeData.Clear()
            OptionData.Clear()
            priority(i) = 0
```

```
            PriorityL1(TreeView1.TopNode.FirstNode, 0)
            Dim k As Integer
            If Ready Then
                For j = 0 To selRow
                    For k = 0 To 5
                        If tempTotal(j, k) <> 0 Then
                            OptionData(j, k) = OptionData(j, k) / tempTotal(j, k)
                        End If
                    Next
                Next
                Dim A As New Matrix()
                A = OptionData.Transpose()
                Dim level As Boolean = True
                Dim checkrow, checkcolumn As Integer
                For checkrow = 0 To selRow
                    For checkcolumn = 0 To 5
                        If NodeData(checkrow, checkcolumn) <> 0 Then
                            level = False
                        End If
                    Next
                Next
                If level Then
                    A = temp.Times(A)
                Else
                    A = NodeData.Times(A)
                    Dim x, y As Integer
                    For x = 0 To selRow
                        For y = 0 To selRow
                            If x = y Then
                                temp0(0, x) = A(x, x)
                            End If
                        Next
                    Next
                    A = temp0.Times(temp)
                End If
                priority(i) = A(0, 0)
                Dim drNode As DataRow
                For Each drNode In objdsAssessment.TblOptions
                    If drNode(0) = selOptions(i) Then
                        drNode(3) = Round(priority(i), 3)
                    End If
                Next
            End If
        Next
        If Ready Then
            sConfirm = Format("The calculation complete sucessfully!")
            MessageBox.Show(sConfirm, "Confirm !", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
            sConfirm = Format("There are not enough information to finish the calculation!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub

    Public Sub LoadDataSet (See same Sub in Module 1)

    Public Sub UpdateDataSet (See same Sub in Module 1)

    Public Sub UpdateDataSource (See same Sub in Module 1)

    Public Sub FillDataSet (See same Sub in Module 1)

    Public Sub LoadTreeView (See same Sub in Module 1)

    Public Sub LoadOptions (See same Sub in Module 1)

    Public Function SearchId (See same Sub in Module 1)

    Public Function SearchModelId (See same Sub in Module 1)

    Private Function SearchTable (See same Sub in Module 1)

    Private Sub ClearEditModel()
        TreeView1.Nodes.Clear()
        Me.TreeView1.Nodes.AddRange(New            System.Windows.Forms.TreeNode()            {New
System.Windows.Forms.TreeNode("Goal")})
        LisOptions.Items.Clear()
        selManager = 0
```

```
        TabAssess.SelectedIndex = 0
    End Sub

    Private Sub ResetDatagrid(ByVal sDatagrid As Datagrid)
        With sDatagrid
            .BackgroundColor = SystemColors.InactiveCaptionText
            .CaptionText = ""
            .CaptionBackColor = SystemColors.ActiveCaption
            .TableStyles.Clear()
            .ResetAlternatingBackColor()
            .ResetBackColor()
            .ResetForeColor()
            .ResetGridLineColor()
            .ResetHeaderBackColor()
            .ResetHeaderFont()
            .ResetHeaderForeColor()
            .ResetSelectionBackColor()
            .ResetSelectionForeColor()
            .ResetText()
            .BorderStyle = BorderStyle.Fixed3D
        End With
    End Sub
```

' Get the datagrid value for pairwise comparison

```
    Private Sub SetPairwise(ByVal sDatagrid As Datagrid)
        With objdsAssessment.TblPairwise
            .DefaultView.RowFilter = "ParentID =" & selParentID
            .DefaultView.AllowNew = False
            sDatagrid.DataSource = .DefaultView
        End With
        With sDatagrid
            .DataSource = objdsAssessment.TblPairwise.DefaultView
            .BackColor = Color.GhostWhite
            .BackgroundColor = Color.GhostWhite
            .CaptionBackColor = Color.RoyalBlue
            .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
            .CaptionText = "Pairwise Comparison"
            .Font = New Font("Tahoma", 12.0!)
            .RowHeadersVisible = False
            .ReadOnly = False
        End With
        Dim TableStyle1 As New DataGridTableStyle()
        With TableStyle1
            .GridLineColor = Color.RoyalBlue
            .MappingName = "TblPairwise"
            .PreferredColumnWidth = 125
            .PreferredRowHeight = 20
            .RowHeadersVisible = False
        End With
        Dim ColumStyle As New DataGridTextBoxColumn()
        With ColumStyle
            .HeaderText = " "
            .MappingName = "LNodeName"
            .ReadOnly = True
            .Alignment = HorizontalAlignment.Center
        End With
        Dim ColumStyle0 As New DataGridTextBoxColumn()
        With ColumStyle0
            .HeaderText = ChildOfNode(0)
            .MappingName = "Child0"
        End With
        Dim ColumStyle1 As New DataGridTextBoxColumn()
        With ColumStyle1
            .HeaderText = ChildOfNode(1)
            .MappingName = "Child1"
        End With
        Dim ColumStyle2 As New DataGridTextBoxColumn()
        With ColumStyle2
            .HeaderText = ChildOfNode(2)
            .MappingName = "Child2"
        End With
        Dim ColumStyle3 As New DataGridTextBoxColumn()
        With ColumStyle3
            .HeaderText = ChildOfNode(3)
            .MappingName = "Child3"
        End With
```

```
        Dim ColumStyle4 As New DataGridTextBoxColumn()
        With ColumStyle4
            .HeaderText = ChildOfNode(4)
            .MappingName = "Child4"
        End With
        Dim ColumStyle5 As New DataGridTextBoxColumn()
        With ColumStyle5
            .HeaderText = ChildOfNode(5)
            .MappingName = "Child5"
        End With
        Dim ColumStyle6 As New DataGridTextBoxColumn()
        With ColumStyle6
            .HeaderText = ChildOfNode(6)
            .MappingName = "Child6"
        End With
        Dim ColumStyle7 As New DataGridTextBoxColumn()
        With ColumStyle7
            .HeaderText = ChildOfNode(7)
            .MappingName = "Child7"
        End With
        Dim ColumStyle8 As New DataGridTextBoxColumn()
        With ColumStyle8
            .HeaderText = ChildOfNode(8)
            .MappingName = "Child8"
        End With
        Dim ColumStyle9 As New DataGridTextBoxColumn()
        With ColumStyle9
            .HeaderText = ChildOfNode(9)
            .MappingName = "Child9"
        End With
        Select Case ChildCount
            Case 0
                sConfirm = Format("There is not enough node for pairwise comparison!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Hand)
            Case 1
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1})
            Case 2
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2})
            Case 3
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3})
            Case 4
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4})
            Case 5
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5})
            Case 6
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6})
            Case 7
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6,
ColumStyle7})
            Case 8
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8})
            Case 9
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9})
        End Select
        sDatagrid.TableStyles.Add(TableStyle1)
        Dim rowNo, colNo As Integer
        For rowNo = 0 To ChildCount
            For colNo = 1 To ChildCount + 1
                If rowNo = colNo - 1 Then
                    objdsAssessment.TblPairwise.DefaultView(rowNo).Item(colNo + 5) = 1
                End If
            Next
        Next
        preRow = 0
        preColumn = 2
    End Sub
```

' Set the datagrid value for scale value datagrid input

```
Private Sub SetDatagridS()
    With objdsAssessment.TblDatagridScale
        .DefaultView.RowFilter = "ModelID =" & selModelID
        .DefaultView.AllowNew = False
        DataScale.DataSource = .DefaultView
    End With
    With DataScale
        .DataSource = objdsAssessment.TblDatagridScale.DefaultView
        .BackgroundColor = Color.GhostWhite
        .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
        .CaptionText = "Scale Value"
        .Font = New Font("Tahoma", 12.0!)
        .RowHeadersVisible = False
    End With
    Dim TableStyle1 As New DataGridTableStyle()
    With TableStyle1
        .BackColor = Color.GhostWhite
        .GridLineColor = Color.RoyalBlue
        .HeaderFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Blue
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen
        .MappingName = "TblDatagridScale"
        .PreferredColumnWidth = 125
        .RowHeadersVisible = False
        .PreferredRowHeight = 20
    End With
    Dim ColumStyle As New DataGridTextBoxColumn()
    With ColumStyle
        .HeaderText = " "
        .MappingName = "NodeName"
        .Alignment = HorizontalAlignment.Left
        .ReadOnly = True
    End With
    Dim ColumStyle0 As New DataGridTextBoxColumn()
    With ColumStyle0
        .HeaderText = ChildOfNode(0)
        .MappingName = "Cover0"
    End With
    Dim ColumStyle1 As New DataGridTextBoxColumn()
    With ColumStyle1
        .HeaderText = ChildOfNode(1)
        .MappingName = "Cover1"
    End With
    Dim ColumStyle2 As New DataGridTextBoxColumn()
    With ColumStyle2
        .HeaderText = ChildOfNode(2)
        .MappingName = "Cover2"
    End With
    Dim ColumStyle3 As New DataGridTextBoxColumn()
    With ColumStyle3
        .HeaderText = ChildOfNode(3)
        .MappingName = "Cover3"
    End With
    Dim ColumStyle4 As New DataGridTextBoxColumn()
    With ColumStyle4
        .HeaderText = ChildOfNode(4)
        .MappingName = "Cover4"
    End With
    Dim ColumStyle5 As New DataGridTextBoxColumn()
    With ColumStyle5
        .HeaderText = ChildOfNode(5)
        .MappingName = "Cover5"
    End With
    Dim ColumStyle6 As New DataGridTextBoxColumn()
    With ColumStyle6
        .HeaderText = ChildOfNode(6)
        .MappingName = "Cover6"
    End With
    Dim ColumStyle7 As New DataGridTextBoxColumn()
    With ColumStyle7
        .HeaderText = ChildOfNode(7)
        .MappingName = "Cover7"
    End With
```

```
        Dim ColumStyle8 As New DataGridTextBoxColumn()
        With ColumStyle8
            .HeaderText = ChildOfNode(8)
            .MappingName = "Cover8"
        End With
        Dim ColumStyle9 As New DataGridTextBoxColumn()
        With ColumStyle9
            .HeaderText = ChildOfNode(9)
            .MappingName = "Cover9"
        End With
        Dim ColumStyle10 As New DataGridTextBoxColumn()
        With ColumStyle10
            .HeaderText = ChildOfNode(10)
            .MappingName = "Cover10"
        End With
        Dim ColumStyle11 As New DataGridTextBoxColumn()
        With ColumStyle11
            .HeaderText = ChildOfNode(11)
            .MappingName = "Cover11"
        End With
        Dim ColumStyle12 As New DataGridTextBoxColumn()
        With ColumStyle12
            .HeaderText = ChildOfNode(12)
            .MappingName = "Cover12"
        End With
        Dim ColumStyle13 As New DataGridTextBoxColumn()
        With ColumStyle13
            .HeaderText = ChildOfNode(13)
            .MappingName = "Cover13"
        End With
        Dim ColumStyle14 As New DataGridTextBoxColumn()
        With ColumStyle14
            .HeaderText = ChildOfNode(14)
            .MappingName = "Cover14"
        End With
        Dim ColumStyle15 As New DataGridTextBoxColumn()
        With ColumStyle15
            .HeaderText = ChildOfNode(15)
            .MappingName = "Cover15"
        End With
        Dim ColumStyle16 As New DataGridTextBoxColumn()
        With ColumStyle16
            .HeaderText = ChildOfNode(16)
            .MappingName = "Cover16"
        End With
        Dim ColumStyle17 As New DataGridTextBoxColumn()
        With ColumStyle17
            .HeaderText = ChildOfNode(17)
            .MappingName = "Cover17"
        End With
        Dim ColumStyle18 As New DataGridTextBoxColumn()
        With ColumStyle18
            .HeaderText = ChildOfNode(18)
            .MappingName = "Cover18"
        End With
        Dim ColumStyle19 As New DataGridTextBoxColumn()
        With ColumStyle19
            .HeaderText = ChildOfNode(19)
            .MappingName = "Cover19"
        End With
        Select Case ChildCount
            Case 1
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1})
            Case 2
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2})
            Case 3
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3})
            Case 4
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4})
            Case 5
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5})
            Case 6
```

```
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6})
            Case 7
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0,  ColumStyle1,  ColumStyle2,  ColumStyle3,  ColumStyle4,  ColumStyle5,  ColumStyle6,
ColumStyle7})
            Case 8
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8})
            Case 9
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9})
            Case 10
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10})
            Case 11
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11})
            Case 12
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11, ColumStyle12})
            Case 13
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11, ColumStyle12, ColumStyle13})
            Case 14
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11, ColumStyle12, ColumStyle13, ColumStyle14})
            Case 15
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15})
            Case 16
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16})
            Case 17
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16, ColumStyle17})
            Case 18
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16, ColumStyle17, ColumStyle18})
            Case 19
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16, ColumStyle17, ColumStyle18, ColumStyle19})
        End Select
        DataScale.TableStyles.Add(TableStyle1)
        preRow = 0
        preColumn = 1
    End Sub

' Set the datagrid value for absolute value datagrid input
    Private Sub SetDatagridA()
        With objdsAssessment.TblDatagrid
            .DefaultView.RowFilter = "ModelID =" & selModelID
            .DefaultView.AllowNew = False
            DataOption.DataSource = .DefaultView
        End With
        With DataOption
            .DataSource = objdsAssessment.TblDatagrid.DefaultView
            .BackgroundColor = Color.GhostWhite
            .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
```

```
        .CaptionText = "Absolute Value"
        .Font = New Font("Tahoma", 12.0!)
        .RowHeadersVisible = False
    End With
    Dim TableStyle1 As New DataGridTableStyle()
    With TableStyle1
        .BackColor = Color.GhostWhite
        .GridLineColor = Color.RoyalBlue
        .HeaderFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Blue
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen
        .MappingName = "TblDatagrid"
        .PreferredColumnWidth = 125
        .RowHeadersVisible = False
        .PreferredRowHeight = 20
    End With
    Dim ColumStyle As New DataGridTextBoxColumn()
    With ColumStyle
        .HeaderText = " "
        .MappingName = "NodeName"
        .ReadOnly = True
        .Alignment = HorizontalAlignment.Left
    End With
    Dim ColumStyle0 As New DataGridTextBoxColumn()
    With ColumStyle0
        .HeaderText = ChildOfNode(0)
        .MappingName = "Cover0"
    End With
    Dim ColumStyle1 As New DataGridTextBoxColumn()
    With ColumStyle1
        .HeaderText = ChildOfNode(1)
        .MappingName = "Cover1"
    End With
    Dim ColumStyle2 As New DataGridTextBoxColumn()
    With ColumStyle2
        .HeaderText = ChildOfNode(2)
        .MappingName = "Cover2"
    End With
    Dim ColumStyle3 As New DataGridTextBoxColumn()
    With ColumStyle3
        .HeaderText = ChildOfNode(3)
        .MappingName = "Cover3"
    End With
    Dim ColumStyle4 As New DataGridTextBoxColumn()
    With ColumStyle4
        .HeaderText = ChildOfNode(4)
        .MappingName = "Cover4"
    End With
    Dim ColumStyle5 As New DataGridTextBoxColumn()
    With ColumStyle5
        .HeaderText = ChildOfNode(5)
        .MappingName = "Cover5"
    End With
    Dim ColumStyle6 As New DataGridTextBoxColumn()
    With ColumStyle6
        .HeaderText = ChildOfNode(6)
        .MappingName = "Cover6"
    End With
    Dim ColumStyle7 As New DataGridTextBoxColumn()
    With ColumStyle7
        .HeaderText = ChildOfNode(7)
        .MappingName = "Cover7"
    End With
    Dim ColumStyle8 As New DataGridTextBoxColumn()
    With ColumStyle8
        .HeaderText = ChildOfNode(8)
        .MappingName = "Cover8"
    End With
    Dim ColumStyle9 As New DataGridTextBoxColumn()
    With ColumStyle9
        .HeaderText = ChildOfNode(9)
        .MappingName = "Cover9"
    End With
    Dim ColumStyle10 As New DataGridTextBoxColumn()
    With ColumStyle10
        .HeaderText = ChildOfNode(10)
```

```
            .MappingName = "Cover10"
        End With
        Dim ColumStyle11 As New DataGridTextBoxColumn()
        With ColumStyle11
            .HeaderText = ChildOfNode(11)
            .MappingName = "Cover11"
        End With
        Dim ColumStyle12 As New DataGridTextBoxColumn()
        With ColumStyle12
            .HeaderText = ChildOfNode(12)
            .MappingName = "Cover12"
        End With
        Dim ColumStyle13 As New DataGridTextBoxColumn()
        With ColumStyle13
            .HeaderText = ChildOfNode(13)
            .MappingName = "Cover13"
        End With
        Dim ColumStyle14 As New DataGridTextBoxColumn()
        With ColumStyle14
            .HeaderText = ChildOfNode(14)
            .MappingName = "Cover14"
        End With
        Dim ColumStyle15 As New DataGridTextBoxColumn()
        With ColumStyle15
            .HeaderText = ChildOfNode(15)
            .MappingName = "Cover15"
        End With
        Dim ColumStyle16 As New DataGridTextBoxColumn()
        With ColumStyle16
            .HeaderText = ChildOfNode(16)
            .MappingName = "Cover16"
        End With
        Dim ColumStyle17 As New DataGridTextBoxColumn()
        With ColumStyle17
            .HeaderText = ChildOfNode(17)
            .MappingName = "Cover17"
        End With
        Dim ColumStyle18 As New DataGridTextBoxColumn()
        With ColumStyle18
            .HeaderText = ChildOfNode(18)
            .MappingName = "Cover18"
        End With
        Dim ColumStyle19 As New DataGridTextBoxColumn()
        With ColumStyle19
            .HeaderText = ChildOfNode(19)
            .MappingName = "Cover19"
        End With
        Select Case ChildCount
            Case 1
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1})
            Case 2
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2})
            Case 3
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3})
            Case 4
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4})
            Case 5
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5})
            Case 6
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6})
            Case 7
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6,
ColumStyle7})
            Case 8
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8})
            Case 9
                TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
```

```
ColumStyle8, ColumStyle9})
            Case 10
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10})
            Case 11
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11})
            Case 12
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11, ColumStyle12})
            Case 13
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11, ColumStyle12, ColumStyle13})
            Case 14
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9, ColumStyle10, ColumStyle11, ColumStyle12, ColumStyle13, ColumStyle14})
            Case 15
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15})
            Case 16
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16})
            Case 17
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16, ColumStyle17})
            Case 18
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16, ColumStyle17, ColumStyle18})
            Case 19
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8,  ColumStyle9,  ColumStyle10,  ColumStyle11,  ColumStyle12,  ColumStyle13,  ColumStyle14,
ColumStyle15, ColumStyle16, ColumStyle17, ColumStyle18, ColumStyle19})
        End Select
        DataOption.TableStyles.Add(TableStyle1)
    End Sub

    Private Sub FindManager()
        Dim frmNew As New DFOmodel_new()
        frmNew.GroupGoal.Text = "Selected Participants"
        frmNew.ComboGoal.Visible = True
        frmNew.ComboGoal.Text = Nothing
        Dim tempModel As DataTable = objdsAssessment.TblManager.Clone
        tempModel.Rows.Clear()
        Dim strExpr As String = Format("ModelID = " & selModelID)
        Dim foundRows() As DataRow
        foundRows = objdsAssessment.TblManagerModel.Select(strExpr)
        Dim i As Integer
        For i = 0 To foundRows.GetUpperBound(0)
            Dim drManager As DataRow
            For Each drManager In objdsAssessment.TblManager
                If drManager.Item("ManagerID") = foundRows(i).Item("ManagerID") Then
                    Dim newRow As DataRow = tempModel.NewRow
                    newRow.Item(0) = drManager.Item(0)
                    newRow.Item(1) = drManager.Item(1)
                    newRow.Item(2) = drManager.Item(2)
                    tempModel.Rows.Add(newRow)
                End If
            Next
        Next i
        frmNew.ComboGoal.DataSource = tempModel
        frmNew.ComboGoal.ValueMember = "ManagerID"
        frmNew.ComboGoal.DisplayMember = "Firstname"
        frmNew.ShowDialog(Me)
```

```
        If frmNew.DialogResult = DialogResult.OK Then
            selManager = frmNew.ComboGoal.SelectedValue
        End If
    End Sub

    Private Sub LoadPairwise()
        Dim drPairwise As DataRow
        ChildCount = 0
        ChildOfNode.Initialize()
        If TreeView1.SelectedNode Is TreeView1.TopNode Then
            selParentID = selModelID
            For Each drPairwise In objdsAssessment.TblPairwise
                If  (drPairwise.Item("ParentID")  =  selParentID)  And  (drPairwise.Item("ManagerID")  =
selManager) Then
                    ChildOfNode(ChildCount) = drPairwise.Item("LNodeName")
                    ChildCount = ChildCount + 1
                    found = True
                End If
            Next
            If found Then
                ChildCount = ChildCount - 1
            Else
                NewPairwise(TreeView1.SelectedNode.FirstNode, selModelID, objdsAssessment.TblNodeL1)
            End If
        End If
        If TreeView1.SelectedNode.Parent Is TreeView1.TopNode Then
            selParentID = SearchId(TreeView1.SelectedNode, selModelID, objdsAssessment.TblNodeL1)
            For Each drPairwise In objdsAssessment.TblPairwise
                If  (drPairwise.Item("ParentID")  =  selParentID)  And  (drPairwise.Item("ManagerID")  =
selManager) Then
                    ChildOfNode(ChildCount) = drPairwise.Item("LNodeName")
                    ChildCount = ChildCount + 1
                    found = True
                End If
            Next
            If found Then
                ChildCount = ChildCount - 1
            Else
                NewPairwise(TreeView1.SelectedNode.FirstNode, selParentID, objdsAssessment.TblNodeL2)
            End If
        End If
    End Sub

    Private Sub NewPairwise(ByVal sNode As TreeNode, ByVal sParentID As Integer, ByVal sTable As DataTable)
        Dim drPairwise As DataRow
        drPairwise = objdsAssessment.TblPairwise.NewRow
        drPairwise.Item("ParentID") = sParentID
        drPairwise.Item("ManagerID") = selManager
        drPairwise.Item("Active") = 1
        drPairwise.Item("ModelID") = selModelID
        drPairwise.Item("LNodeID") = SearchId(sNode, sParentID, sTable)
        drPairwise.Item("LNodeName") = sNode.Text
        Dim i As Integer
        For i = 6 To 15
            drPairwise.Item(i) = 0
        Next
        objdsAssessment.TblPairwise.Rows.Add(drPairwise)
        ChildOfNode(ChildCount) = sNode.Text
        If Not (sNode.NextNode Is Nothing) Then
            ChildCount = ChildCount + 1
            NewPairwise(sNode.NextNode, sParentID, sTable)
        End If
    End Sub

    Private Sub LoadDatagrid(ByVal sTable As DataTable)
        Dim drDatagrid As DataRow
        ChildCount = 0
        ChildOfNode.Initialize()
        found = False
        For Each drDatagrid In sTable.Rows
            If (drDatagrid.Item("ModelID") = selModelID) And (drDatagrid.Item("ManagerID") = selManager)
Then
                found = True
            End If
        Next
        If Not found Then
            Dim drOption As DataRow
```

```
            For Each drOption In objdsAssessment.TblOptions
                If drOption.Item("ModelID") = selModelID Then
                    drDatagrid = sTable.NewRow
                    drDatagrid.Item("OptionID") = drOption.Item("OptionID")
                    drDatagrid.Item("NodeName") = drOption.Item("NodeName")
                    drDatagrid.Item("ModelID") = selModelID
                    drDatagrid.Item("ManagerID") = selManager
                    drDatagrid.Item("Active") = 1
                    Dim i As Integer
                    For i = 5 To 24
                        drDatagrid.Item(i) = 0
                    Next
                    sTable.Rows.Add(drDatagrid)
                End If
            Next
        End If
        NewDatagrid(TreeView1.TopNode.FirstNode)
        ChildCount = ChildCount - 1
    End Sub

    Private Sub NewDatagrid(ByVal sNode As TreeNode)
        If sNode.FirstNode Is Nothing Then
            ChildOfNode(ChildCount) = sNode.Text
            ChildCount = ChildCount + 1
            If Not (sNode.NextNode Is Nothing) Then
                NewDatagrid(sNode.NextNode)
            End If
        Else
            Dim aNode As TreeNode
            For Each aNode In sNode.Nodes
                ChildOfNode(ChildCount) = sNode.Text & "-" & aNode.Text
                ChildCount = ChildCount + 1
            Next
            If Not (sNode.NextNode Is Nothing) Then
                NewDatagrid(sNode.NextNode)
            End If
        End If
    End Sub

    Function CreateDataSource() As ICollection
        Dim dt As New DataTable()
        Dim columnName As String
        Dim strExpr As String = Format("SchemeID = " & selScheme)
        If selScheme = 0 Then
            sConfirm = Format("You need to set up the rating scale for data input!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Else
            Dim IntensityView As DataView = New DataView(objdsAssessment.TblIntensity)
            IntensityView.RowFilter = strExpr
            IntensityView.Sort = "ShowID"
            Dim i As Integer
            For i = 0 To IntensityView.Count - 1
                columnName = "         " & IntensityView.Item(i).Item("ShowID")
                dt.Columns.Add(New DataColumn(columnName, GetType(Object)))
                ShowID(i) = IntensityView.Item(i).Item("ShowID")
                ShowValue(i) = IntensityView.Item(i).Item("Priority")
            Next
            MaxValue = ShowID(IntensityView.Count - 1)
            MinValue = ShowID(0)
            Dim dr As DataRow = dt.NewRow()
            For i = 0 To IntensityView.Count - 1
                dr(i)    =    "        "    &    IntensityView.Item(i).Item("RateName")    &    "    ("    &
IntensityView.Item(i).Item("Priority") & ")"
            Next
            dt.Rows.Add(dr)
            Dim dv As New DataView(dt)
            Return dv
        End If
    End Function

    Private Sub CalPairwise(ByVal sNode As TreeNode, ByVal sParentID As Integer)
        If Ready Then
            Dim i, j, selParent As Integer
            Dim aNode As TreeNode
            For Each aNode In sNode.Nodes
                If Not (aNode.FirstNode Is Nothing) Then
                    selParent = SearchId(aNode, sParentID, objdsAssessment.TblNodeL1)
```

```
                CalPairwise(aNode, selParent)
            End If
        Next
        Dim strExpr As String
        strExpr = Format("ParentID = " & sParentID)
        Dim foundRows() As DataRow
        foundRows = objdsAssessment.TblPairwise.Select(strExpr)
        index = foundRows.GetUpperBound(0)
        If index = 0 Then
            Ready = False
        Else
            Dim Id(index + 1) As Integer
            Dim A As New Matrix()
            A.Size(index + 1, index + 1)
            For i = 0 To index
                Id(i) = foundRows(i).Item("LNodeID")
                For j = 0 To index
                    A(i, j) = foundRows(i).Item(j + 6)
                    If A(i, j) = 0 Then
                        Ready = False
                    End If
                Next j
            Next i
            If Ready Then
                Dim X As New Matrix()
                Dim Priority(index + 1) As Double
                Dim Lambda As Double = 0
                Dim inconsistency As Decimal = -1
                Dim k As Integer
                Dim Vr As New Matrix()
                Dim Vi As New Matrix()
                Dim Er As New Matrix()
                Dim Ei As New Matrix()
                A.Eigen(Vr, Vi, Er, Ei)
                For k = 0 To index
                    If Vi(k, 0) = 0 And Lambda < Vr(k, 0) Then
                        Lambda = Vr(k, 0)
                        X = Er.ColVector(k)
                        inconsistency = Abs(Lambda - index - 1) / index
                    End If
                Next
                If Not (inconsistency > 0 And inconsistency < 0.1) Then
                    sConfirm = Format("You need to adjust your pairwise comparison to get the right
priorities!")
                    MessageBox.Show(sConfirm,       "Warning       !",       MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                End If
                Dim total As Double = 0
                For k = 0 To index
                    total = total + X(k, 0)
                Next
                For k = 0 To index
                    Priority(k) = X(k, 0) / total
                Next
                For i = 0 To index
                    Dim drNode As DataRow
                    If Id(i) < 1000000 Then
                        For Each drNode In objdsAssessment.TblNodeL1
                            If drNode(0) = Id(i) Then
                                drNode(4) = Round(Priority(i), 3)
                            End If
                        Next
                    Else
                        For Each drNode In objdsAssessment.TblNodeL2
                            If drNode(0) = Id(i) Then
                                drNode(4) = Round(Priority(i), 3)
                            End If
                        Next
                    End If
                Next
                sNode.ImageIndex = 12
                Dim theNode As TreeNode
                For Each theNode In sNode.Nodes
                    theNode.ImageIndex = 12
                Next
            End If
        End If
```

```
        End If
    End Sub

    Private Sub PriorityL1(ByVal sNode As TreeNode, ByVal k As Integer)
        If Ready Then
            Dim sNodeID As Integer = SearchId(sNode, selModelID, objdsAssessment.TblNodeL1)
            Dim drNode As DataRow
            For Each drNode In objdsAssessment.TblNodeL1.Rows
                If drNode("NodeID") = sNodeID Then
                    temp(k, 0) = 0
                    temp(k, 0) = drNode("Priority")
                    If temp(k, 0) = 0 Then
                        Ready = False
                    End If
                End If
            Next
            If Not (sNode.FirstNode Is Nothing) Then
                PriorityL2(sNode.FirstNode, sNodeID, k, 0)
            Else
                OptionData(k, 0) = DataOption(curRow, curColumn)
            End If
            If Not (sNode.NextNode Is Nothing) Then
                curColumn += 1
                k += 1
                PriorityL1(sNode.NextNode, k)
            End If
        End If
    End Sub

    Private Sub PriorityL2(ByVal sNode As TreeNode, ByVal sParent As Integer, ByVal row As Integer, ByVal
column As Integer)
        If Ready Then
            Dim sNodeID As Integer = SearchId(sNode, sParent, objdsAssessment.TblNodeL2)
            Dim drNode As DataRow
            For Each drNode In objdsAssessment.TblNodeL2.Rows
                If drNode("NodeID") = sNodeID Then
                    NodeData(row, column) = 0
                    NodeData(row, column) = drNode("Priority")
                    OptionData(row, column) = DataOption(curRow, curColumn)
                    If NodeData(row, column) = 0 Then
                        Ready = False
                    End If
                End If
            Next
            If Not (sNode.NextNode Is Nothing) Then
                curColumn += 1
                column += 1
                PriorityL2(sNode.NextNode, sParent, row, column)
            End If
        End If
    End Sub

    Private Sub CheckChange()
        Dim objChanges As DFOdemo.dsAssessment = New DFOdemo.dsAssessment()
        objChanges = CType(objdsAssessment.GetChanges, DFOdemo.dsAssessment)
        If (Not (objChanges) Is Nothing) Then
            Dim sConfirm As String = String.Format("Do you want to save the change?")
            selResult = MessageBox.Show(sConfirm, "Comfirm Save", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
            If selResult = DialogResult.Yes Then
                Try
                    Me.UpdateDataSet()
                Catch eUpdate As System.Exception
                    System.Windows.Forms.MessageBox.Show(eUpdate.Message)
                End Try
            Else
                objdsAssessment.RejectChanges()
            End If
        End If
    End Sub
#End Region
```

' The following code handle the event raised by certain operation other than selecting the menu such as tab switch, data input and scale setup.

#Region "Event Handle"

```
    Private     Sub     DFOassess_Closing(ByVal     sender     As     Object,     ByVal     e     As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
        CheckChange()
    End Sub

    Private Sub TabAssess_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TabAssess.SelectedIndexChanged
        Select Case TabAssess.SelectedIndex
            Case 0
                MenuPairwise.Enabled = False
                MenuDatagrid.Enabled = False
                ToolBar1.Buttons(2).Visible = False
                ToolBar1.Buttons(3).Visible = False
                ToolBar1.Buttons(4).Visible = False
            Case 1
                If TreeView1.SelectedNode Is Nothing Then
                    sConfirm = Format("You need to select an node for pairwise comparison!")
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                    TabAssess.SelectedIndex = 0
                    TreeView1.SelectedNode = TreeView1.TopNode
                Else
                    If (TreeView1.SelectedNode Is TreeView1.TopNode) Or (TreeView1.SelectedNode.Parent Is
TreeView1.TopNode) Then
                        MenuPairwise.Enabled = True
                        MenuDatagrid.Enabled = False
                        ToolBar1.Buttons(2).Visible = True
                        ToolBar1.Buttons(3).Visible = True
                        ToolBar1.Buttons(4).Visible = False
                        If selManager = 0 Then
                            FindManager()
                            If selManager <> 0 Then
                                LoadPairwise()
                                ResetDatagrid(VerbalDataGrid)
                                SetPairwise(VerbalDataGrid)
                                VerbalParent.Text = TreeView1.SelectedNode.Text
                                found = False
                            Else
                                TabAssess.SelectedIndex = 0
                                TreeView1.SelectedNode = TreeView1.TopNode
                            End If
                        Else
                            LoadPairwise()
                            ResetDatagrid(VerbalDataGrid)
                            SetPairwise(VerbalDataGrid)
                            VerbalParent.Text = TreeView1.SelectedNode.Text
                            found = False
                        End If
                    Else
                        sConfirm = Format("You can't make pairwise comparison with respect to the lowest
level element!")
                        MessageBox.Show(sConfirm,       "Warning       !",       MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                        TabAssess.SelectedIndex = 0
                        TreeView1.SelectedNode = TreeView1.TopNode
                    End If
                End If
            Case 2
                If TreeView1.SelectedNode Is Nothing Then
                    sConfirm = Format("You need to select an node for pairwise comparison!")
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                    TabAssess.SelectedIndex = 0
                    TreeView1.SelectedNode = TreeView1.TopNode
                Else
                    If (TreeView1.SelectedNode Is TreeView1.TopNode) Or (TreeView1.SelectedNode.Parent Is
TreeView1.TopNode) Then
                        MenuPairwise.Enabled = True
                        MenuDatagrid.Enabled = False
                        ToolBar1.Buttons(2).Visible = True
                        ToolBar1.Buttons(3).Visible = True
                        ToolBar1.Buttons(4).Visible = False
                        If selManager = 0 Then
                            FindManager()
                            If selManager <> 0 Then
                                LoadPairwise()
```

```
                        ResetDatagrid(GraphicDataGrid)
                        GraphicParent.Text = TreeView1.SelectedNode.Text
                        SetPairwise(GraphicDataGrid)
                        found = False
                    Else
                        TabAssess.SelectedIndex = 0
                        TreeView1.SelectedNode = TreeView1.TopNode
                    End If
                Else
                    LoadPairwise()
                    ResetDatagrid(GraphicDataGrid)
                    GraphicParent.Text = TreeView1.SelectedNode.Text
                    SetPairwise(GraphicDataGrid)
                    found = False
                End If
            Else
                sConfirm = Format("You can't make pairwise comparison with respect to the lowest
level element!")
                MessageBox.Show(sConfirm,        "Warning       !",        MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                TabAssess.SelectedIndex = 0
                TreeView1.SelectedNode = TreeView1.TopNode
            End If
        End If
    Case 3
        MenuPairwise.Enabled = False
        MenuDatagrid.Enabled = True
        ToolBar1.Buttons(2).Visible = True
        ToolBar1.Buttons(3).Visible = False
        ToolBar1.Buttons(4).Visible = True
        With DataRate
            .BackgroundColor = Color.GhostWhite
            .GridLineColor = Color.RoyalBlue
            .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
            .CaptionText = "Scale of Rating"
            .Font = New Font("Tahoma", 12.0!)
        End With
        If selManager = 0 Then
            FindManager()
            If selManager <> 0 Then
                With DataRate
                    .DataSource = CreateDataSource()
                End With
                LoadDatagrid(objdsAssessment.TblDatagrid)
                ResetDatagrid(DataOption)
                SetDatagridA()
                LoadDatagrid(objdsAssessment.TblDatagridScale)
                ResetDatagrid(DataScale)
                SetDatagridS()
            Else
                TabAssess.SelectedIndex = 0
                TreeView1.SelectedNode = TreeView1.TopNode
            End If
        Else
            With DataRate
                .DataSource = CreateDataSource()
            End With
            LoadDatagrid(objdsAssessment.TblDatagrid)
            ResetDatagrid(DataOption)
            SetDatagridA()
            LoadDatagrid(objdsAssessment.TblDatagridScale)
            ResetDatagrid(DataScale)
            SetDatagridS()
        End If
    End Select
End Sub

Private  Sub  VerbalDataGrid_CurrentCellChanged(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles VerbalDataGrid.CurrentCellChanged
    curRow = VerbalDataGrid.CurrentCell.RowNumber
    curColumn = VerbalDataGrid.CurrentCell.ColumnNumber
    If VerbalDataGrid(preRow, preColumn) <> 0 Then
        Dim selValue As Decimal = 1 / VerbalDataGrid(preRow, preColumn)
        objdsAssessment.TblPairwise.DefaultView(preColumn - 1).Item(preRow + 1 + 5) = Round(selvalue,
3)
    End If
    LVerbalNode.Text = VerbalDataGrid.Item(curRow, 0)
```

```
        RVerbalNode.Text = ChildOfNode(curColumn - 1)
        If Not (VerbalDataGrid(curRow, curColumn) = 0) Then
            Select Case VerbalDataGrid(curRow, curColumn)
                Case Is = 1
                    TrackBar.Value = 8
                Case Is < 1
                    Dim selValue As Integer = Round(1 / VerbalDataGrid(curRow, curColumn))
                    If selValue = 1 Then
                        TrackBar.Value = 8
                    Else
                        TrackBar.Value = 9 - selvalue
                    End If
                Case Is > 1
                    Dim selValue As Integer = Round(VerbalDataGrid(curRow, curColumn))
                    If selValue = 1 Then
                        TrackBar.Value = 8
                    Else
                        TrackBar.Value = selvalue + 7
                    End If
            End Select
        Else
            TrackBar.Value = 8
        End If
        preRow = curRow
        preColumn = curColumn
    End Sub

    Private Sub TrackBar_MouseUp(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles TrackBar.MouseUp
        curRow = VerbalDataGrid.CurrentCell.RowNumber
        curColumn = VerbalDataGrid.CurrentCell.ColumnNumber
        Dim diaRow As Integer = curColumn - 1
        Dim diaColumn As Integer = curRow + 1
        Dim selValue As Decimal
        If curRow = curColumn - 1 Then
            VerbalDataGrid(curRow, curColumn) = 1
            TrackBar.Value = 8
        Else
            Select Case TrackBar.Value
                Case Is = 8
                    VerbalDataGrid(curRow, curColumn) = 1
                Case Is < 8
                    selValue = 1 / (9 - TrackBar.Value)
                    VerbalDataGrid(curRow, curColumn) = Round(selValue, 3)
                    objdsAssessment.TblPairwise.DefaultView(diaRow).Item(diaColumn + 5) = 9 -
TrackBar.Value
                Case Is > 8
                    selValue = 1 / (TrackBar.Value - 7)
                    VerbalDataGrid(curRow, curColumn) = TrackBar.Value - 7
                    objdsAssessment.TblPairwise.DefaultView(diaRow).Item(diaColumn + 5) = Round(selValue,
3)
            End Select
        End If
    End Sub

    Private Sub GraphicDataGrid_CurrentCellChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles GraphicDataGrid.CurrentCellChanged
        curRow = GraphicDataGrid.CurrentCell.RowNumber
        curColumn = GraphicDataGrid.CurrentCell.ColumnNumber
        If GraphicDataGrid(preRow, preColumn) <> 0 Then
            Dim selValue As Decimal = 1 / GraphicDataGrid(preRow, preColumn)
            objdsAssessment.TblPairwise.DefaultView(preColumn - 1).Item(preRow + 1 + 5) = Round(selvalue,
3)
        End If
        LGraphicNode.Text = GraphicDataGrid.Item(curRow, 0)
        RGraphicNode.Text = ChildOfNode(curColumn - 1)
        If Not (GraphicDataGrid(curRow, curColumn) = 0) Then
            Select Case GraphicDataGrid(curRow, curColumn)
                Case Is = 1
                    LPercent.PortionValue = 50
                    RPercent.PortionValue = 50
                    PiePercent.PortionValue = 50
                Case Is < 1
                    Dim selValue As Decimal = GraphicDataGrid(curRow, curColumn) * 100
                    selvalue = Round(selvalue / (GraphicDataGrid(curRow, curColumn) + 1), 2)
                    LPercent.PortionValue = selvalue
                    RPercent.PortionValue = 100 - selvalue
```

```
                PiePercent.PortionValue = selvalue
            Case Is > 1
                Dim selValue As Decimal = GraphicDataGrid(curRow, curColumn) * 100
                selvalue = Round(selvalue / (GraphicDataGrid(curRow, curColumn) + 1), 2)
                LPercent.PortionValue = selvalue
                RPercent.PortionValue = 100 - selvalue
                PiePercent.PortionValue = selvalue
        End Select
    Else
        LPercent.PortionValue = 50
        RPercent.PortionValue = 50
        PiePercent.PortionValue = 50
    End If
    preRow = curRow
    preColumn = curColumn
End Sub

Private Sub ToolBar1_ButtonClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar1.ButtonClick
    Select Case ToolBar1.Buttons.IndexOf(e.Button)
        Case 0
            MenuOpenModel()
        Case 1
            Try
                Me.UpdateDataSet()
            Catch eUpdate As System.Exception
                System.Windows.Forms.MessageBox.Show(eUpdate.Message)
            End Try
        Case 3
            Ready = True
            CalPairwise(TreeView1.TopNode, selModelID)
            If Ready Then
                sConfirm = Format("The calculation complete sucessfully!")
                MessageBox.Show(sConfirm, "Confirm !", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Else
                sConfirm = Format("There are not enough information to finish the calculation!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            End If
        Case 4
            MenuCalDatagrid()
        Case 6
            Dim frmAbout As New DFO_About()
            frmAbout.ShowDialog(Me)
            frmAbout.Dispose()
    End Select

End Sub

Private Sub LPercent_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
LPercent.Change
    If LGraphicNode.Text = " " Then
        sConfirm = Format("You need to select a cell in the datagrid for your input!")
        MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        LPercent.PortionValue = 50
    Else
        RPercent.PortionValue = 100 - LPercent.PortionValue
        PiePercent.PortionValue = LPercent.PortionValue
        curRow = GraphicDataGrid.CurrentCell.RowNumber
        curColumn = GraphicDataGrid.CurrentCell.ColumnNumber
        GraphicDataGrid(curRow, curColumn) = Round(LPercent.PortionValue / RPercent.PortionValue, 3)
        Dim selValue As Decimal = 1 / GraphicDataGrid(curRow, curColumn)
        objdsAssessment.TblPairwise.DefaultView(curColumn - 1).Item(curRow + 1 + 5) = Round(selValue,
3)
    End If
End Sub

Private Sub RPercent_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RPercent.Change
    If LGraphicNode.Text = " " Then
        sConfirm = Format("You need to select a cell in the datagrid for your input!")
        MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        RPercent.PortionValue = 50
    Else
        LPercent.PortionValue = 100 - RPercent.PortionValue
        PiePercent.PortionValue = LPercent.PortionValue
        curRow = GraphicDataGrid.CurrentCell.RowNumber
```

```
            curColumn = GraphicDataGrid.CurrentCell.ColumnNumber
            GraphicDataGrid(curRow, curColumn) = Round(LPercent.PortionValue / RPercent.PortionValue, 3)
            Dim selValue As Decimal = 1 / GraphicDataGrid(curRow, curColumn)
            objdsAssessment.TblPairwise.DefaultView(curColumn - 1).Item(curRow + 1 + 5) = Round(selValue,
3)
        End If
    End Sub

    Private Sub PiePercent_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PiePercent.Change
        If LGraphicNode.Text = " " Then
            sConfirm = Format("You need to select a cell in the datagrid for your input!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            PiePercent.PortionValue = 50
        Else
            LPercent.PortionValue = PiePercent.PortionValue
            RPercent.PortionValue = 100 - LPercent.PortionValue
            curRow = GraphicDataGrid.CurrentCell.RowNumber
            curColumn = GraphicDataGrid.CurrentCell.ColumnNumber
            GraphicDataGrid(curRow, curColumn) = Round(LPercent.PortionValue / RPercent.PortionValue, 3)
            Dim selValue As Decimal = 1 / GraphicDataGrid(curRow, curColumn)
            objdsAssessment.TblPairwise.DefaultView(curColumn - 1).Item(curRow + 1 + 5) = Round(selValue,
3)
        End If
    End Sub

    Private Sub DataScale_CurrentCellChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles
DataScale.CurrentCellChanged
        If preColumn > 0 Then
            If DataScale(preRow, preColumn) <> 0 Then
                Dim i As Integer
                If MaxValue < DataScale(preRow, preColumn) Or MinValue > DataScale(preRow, preColumn) Then
                    sConfirm = Format("Input data is out of range of rating scheme!")
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                    DataScale(preRow, preColumn) = 0
                Else
                    For i = 0 To 9
                        If ShowID(i) = DataScale(preRow, preColumn) Then
                            DataOption(preRow, preColumn) = ShowValue(i)
                        End If
                    Next
                End If
            End If
        End If
        preRow = DataScale.CurrentCell.RowNumber
        preColumn = DataScale.CurrentCell.ColumnNumber
    End Sub

    Private Sub BtnSetup_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnSetup.Click
        Dim frmSetup As New DFOrate_Setup()
        Me.Hide()
        frmSetup.ShowDialog(Me)
        Me.Show()
        selScheme = frmSetup.ComboRate.SelectedValue
        Dim drModel As DataRow
        For Each drModel In objdsAssessment.TblModel
            If drModel.Item("ModelID") = selModelID Then
                drModel.Item("SchemeID") = selScheme
            End If
        Next
        With DataRate
            .DataSource = CreateDataSource()
        End With
    End Sub
#End Region
End Class
```

Module 2.2 Rating Scale Setup

This module is designed to set up the rating scale for datagrid evaluation purpose.

```
Imports System.Math
Imports BluebitMatrix30
```

```
Public Class DFOrate_Setup
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim sConfirm, selResult As String
    Dim ready As Boolean
    Dim selScheme, ChildCount, preRow, preColumn, curRow, curColumn As Integer
    Dim ChildOfNode(20) As String
    Private Sub DFOrate_Setup_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        ComboRate.DataSource = objdsAssessment.TblScheme
        ComboRate.DisplayMember = "SchemeName"
        ComboRate.ValueMember = "SchemeID"
    End Sub
```

#Region "Menu"

```
    Private Sub MenuAbout_Click (See same Sub in Module 1)

    Private Sub MenuSave_Click (See same Sub in Module 1)

    Private Sub MenuExit_Click (See same Sub in Module 1)

    Private Sub MenuDel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuDel.Click
        If Not selScheme > 0 Then
            sConfirm = Format("No rating scheme is selected!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Else
            sConfirm = String.Format("The selected rating scheme will be deleted permanently, continue?")
            selResult = MessageBox.Show(sConfirm, "Comfirm Delete", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
            If selResult = DialogResult.OK Then
                Dim strExpr As String
                strExpr = Format("SchemeID = " & selScheme)
                Dim foundRows() As DataRow
                foundRows = objdsAssessment.TblIntensity.Select(strExpr)
                Dim i As Integer
                For i = 0 To foundRows.GetUpperBound(0)
                    foundRows(i).Delete()
                Next i
                foundRows = objdsAssessment.TblRating.Select(strExpr)
                For i = 0 To foundRows.GetUpperBound(0)
                    foundRows(i).Delete()
                Next i
                foundRows = objdsAssessment.TblScheme.Select(strExpr)
                For i = 0 To foundRows.GetUpperBound(0)
                    foundRows(i).Delete()
                Next i
                foundRows = objdsAssessment.TblModel.Select(strExpr)
                For i = 0 To foundRows.GetUpperBound(0)
                    foundRows(i).Item("SchemeID") = 0
                Next i
                ClearEdit()
            End If
        End If
    End Sub
#End Region
```

#Region "Sub"

```
    Public Sub LoadDataSet (See same Sub in Module 1)

    Private Sub Row_Changed(ByVal sender As Object, ByVal e As DataRowChangeEventArgs)
        If e.Row("SchemeID") Is DBNull.Value Then
            e.Row("SchemeID") = selScheme
        End If
    End Sub
```

```
Public Sub UpdateDataSet (See same Sub in Module 1)

Public Sub UpdateDataSource(ByVal ChangedRows As DFOdemo.dsAssessment)
    Try
        If (Not (ChangedRows) Is Nothing) Then
            Me.DbConnection.Open()
            daScheme.Update(ChangedRows)
            daRating.Update(ChangedRows)
            daIntensity.Update(ChangedRows)
        End If
    Catch updateException As System.Exception
        Throw updateException
    Finally
        Me.DbConnection.Close()
    End Try
End Sub

Public Sub FillDataSet(ByVal dataSet As DFOdemo.dsAssessment)
    dataSet.EnforceConstraints = False
    Try
        Me.DbConnection.Open()
        Me.daScheme.Fill(dataSet)
        Me.daRating.Fill(dataSet)
        Me.daIntensity.Fill(dataSet)
    Catch fillException As System.Exception
        Throw fillException
    Finally
        dataSet.EnforceConstraints = True
        Me.DbConnection.Close()
    End Try
End Sub

Private Sub ResetDatagrid(See same Sub in Module 2)
```

' Set the datagrid value for selected scale scheme

```
Private Sub SetIntensity()
    With objdsAssessment.TblIntensity
        .DefaultView.RowFilter = "SchemeID =" & selScheme
    End With
    Dim selSchemeName As String
    If RadButNew.Checked Then
        selSchemeName = TextRate.Text
    Else
        selSchemeName = ComboRate.Text
    End If
    With GridIntensity
        .DataSource = objdsAssessment.TblIntensity.DefaultView
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.GhostWhite
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
        .CaptionText = selSchemeName
        .Font = New Font("Tahoma", 12.0!)
        .RowHeadersVisible = False
        .ReadOnly = False
    End With
    Dim TableStyle1 As New DataGridTableStyle()
    With TableStyle1
        .GridLineColor = Color.RoyalBlue
        .MappingName = "TblIntensity"
        .PreferredColumnWidth = 150
        .PreferredRowHeight = 20
        .RowHeadersVisible = False
    End With
    Dim ColumStyle As New DataGridTextBoxColumn()
    With ColumStyle
        .HeaderText = " Rate Name"
        .MappingName = "RateName"
        .Alignment = HorizontalAlignment.Center
        .Width = 250
    End With
    Dim ColumStyle0 As New DataGridTextBoxColumn()
    With ColumStyle0
        .HeaderText = "Priority"
        .MappingName = "Priority"
        .Alignment = HorizontalAlignment.Center
```

```
        End With
        TableStyle1.GridColumnStyles.AddRange(New DataGridColumnStyle() {ColumStyle, ColumStyle0})
        GridIntensity.TableStyles.Add(TableStyle1)
    End Sub
```

' Load the datagrid for selected scale scheme datagrid input

```
    Private Sub LoadRating()
        Dim drPairwise As DataRow
        ChildCount = 0
        ChildOfNode.Initialize()
        Dim i As Integer
        ready = False
        For i = 0 To objdsAssessment.TblIntensity.DefaultView.Count - 1
            For Each drPairwise In objdsAssessment.TblRating
                If (drPairwise.Item("SchemeID") = selScheme) And (drPairwise.Item("RateID") =
objdsAssessment.TblIntensity.DefaultView.Item(i).Item("RateID")) Then
                    drPairwise.Item("RateName") =
objdsAssessment.TblIntensity.DefaultView.Item(i).Item("RateName")
                    ChildOfNode(ChildCount) =
objdsAssessment.TblIntensity.DefaultView.Item(i).Item("RateName")
                    ChildCount = ChildCount + 1
                    ready = True
                End If
            Next
            If Not ready Then
                Dim drRating As DataRow
                drRating = objdsAssessment.TblRating.NewRow
                drRating.Item("SchemeID") = selScheme
                drRating.Item("RateID") =
objdsAssessment.TblIntensity.DefaultView.Item(i).Item("RateID")
                drRating.Item("RateName") =
objdsAssessment.TblIntensity.DefaultView.Item(i).Item("RateName")
                Dim j As Integer
                For j = 3 To 12
                    drRating.Item(j) = 0
                Next
                objdsAssessment.TblRating.Rows.Add(drRating)
                ChildOfNode(ChildCount) =
objdsAssessment.TblIntensity.DefaultView.Item(i).Item("RateName")
                ChildCount = ChildCount + 1
            End If
        Next
        ChildCount = ChildCount - 1
    End Sub
```

' Set the datagrid input for selected scale scheme

```
    Private Sub SetRating()
        With objdsAssessment.TblRating
            .DefaultView.RowFilter = "SchemeID =" & selScheme
            .DefaultView.AllowNew = False
            GridRating.DataSource = .DefaultView
        End With
        With GridRating
            .DataSource = objdsAssessment.TblRating.DefaultView
            .BackColor = Color.GhostWhite
            .BackgroundColor = Color.GhostWhite
            .CaptionBackColor = Color.RoyalBlue
            .CaptionFont = New Font("Tahoma", 12.0!, FontStyle.Bold)
            .CaptionText = "Pairwise Comparison"
            .Font = New Font("Tahoma", 12.0!)
            .RowHeadersVisible = False
        End With
        Dim TableStyle1 As New DataGridTableStyle()
        With TableStyle1
            .GridLineColor = Color.RoyalBlue
            .MappingName = "TblRating"
            .PreferredColumnWidth = 125
            .PreferredRowHeight = 20
            .RowHeadersVisible = False
        End With
        Dim ColumStyle As New DataGridTextBoxColumn()
        With ColumStyle
            .HeaderText = " "
            .MappingName = "RateName"
            .ReadOnly = True
            .Alignment = HorizontalAlignment.Center
```

```
        End With
        Dim ColumStyle0 As New DataGridTextBoxColumn()
        With ColumStyle0
            .HeaderText = ChildOfNode(0)
            .MappingName = "Child0"
        End With
        Dim ColumStyle1 As New DataGridTextBoxColumn()
        With ColumStyle1
            .HeaderText = ChildOfNode(1)
            .MappingName = "Child1"
        End With
        Dim ColumStyle2 As New DataGridTextBoxColumn()
        With ColumStyle2
            .HeaderText = ChildOfNode(2)
            .MappingName = "Child2"
        End With
        Dim ColumStyle3 As New DataGridTextBoxColumn()
        With ColumStyle3
            .HeaderText = ChildOfNode(3)
            .MappingName = "Child3"
        End With
        Dim ColumStyle4 As New DataGridTextBoxColumn()
        With ColumStyle4
            .HeaderText = ChildOfNode(4)
            .MappingName = "Child4"
        End With
        Dim ColumStyle5 As New DataGridTextBoxColumn()
        With ColumStyle5
            .HeaderText = ChildOfNode(5)
            .MappingName = "Child5"
        End With
        Dim ColumStyle6 As New DataGridTextBoxColumn()
        With ColumStyle6
            .HeaderText = ChildOfNode(6)
            .MappingName = "Child6"
        End With
        Dim ColumStyle7 As New DataGridTextBoxColumn()
        With ColumStyle7
            .HeaderText = ChildOfNode(7)
            .MappingName = "Child7"
        End With
        Dim ColumStyle8 As New DataGridTextBoxColumn()
        With ColumStyle8
            .HeaderText = ChildOfNode(8)
            .MappingName = "Child8"
        End With
        Dim ColumStyle9 As New DataGridTextBoxColumn()
        With ColumStyle9
            .HeaderText = ChildOfNode(9)
            .MappingName = "Child9"
        End With
        Select Case ChildCount
            Case 0
                sConfirm = Format("There is not enough node for pairwise comparison!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Hand)
            Case 1
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1})
            Case 2
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2})
            Case 3
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3})
            Case 4
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4})
            Case 5
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5})
            Case 6
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6})
            Case 7
                TableStyle1.GridColumnStyles.AddRange(New      DataGridColumnStyle()      {ColumStyle,
ColumStyle0,   ColumStyle1,   ColumStyle2,   ColumStyle3,   ColumStyle4,   ColumStyle5,   ColumStyle6,
ColumStyle7})
```

```
            Case 8
                TableStyle1.GridColumnStyles.AddRange(New       DataGridColumnStyle()       {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8})
            Case 9
                TableStyle1.GridColumnStyles.AddRange(New       DataGridColumnStyle()       {ColumStyle,
ColumStyle0, ColumStyle1, ColumStyle2, ColumStyle3, ColumStyle4, ColumStyle5, ColumStyle6, ColumStyle7,
ColumStyle8, ColumStyle9})
        End Select
        GridRating.TableStyles.Add(TableStyle1)
        Dim rowNo, colNo As Integer
        For rowNo = 0 To ChildCount
            For colNo = 1 To ChildCount + 1
                If rowNo = colNo - 1 Then
                    objdsAssessment.TblRating.DefaultView(rowNo).Item(colNo + 2) = 1
                End If
            Next
        Next
        preRow = 0
        preColumn = 2
    End Sub

    Private Sub CalPairwise()
        Dim i, j As Integer
        Dim Id(GridRating.VisibleRowCount) As Integer
        Dim A As New Matrix()
        A.Size(GridRating.VisibleRowCount, GridRating.VisibleColumnCount - 1)
        For i = 0 To GridRating.VisibleRowCount - 1
            For j = 0 To GridRating.VisibleColumnCount - 2
                A(i, j) = GridRating.Item(i, j + 1)
                Id(i) = objdsAssessment.TblRating.DefaultView.Item(i).Item("RateID")
            Next j
        Next i
        Dim X As New Matrix()
        Dim Priority(GridRating.VisibleRowCount) As Double
        Dim Lambda As Double
        Dim k As Integer
        Dim Vr As New Matrix()
        Dim Vi As New Matrix()
        Dim Er As New Matrix()
        Dim Ei As New Matrix()
        A.Eigen(Vr, Vi, Er, Ei)
        For k = 0 To GridRating.VisibleRowCount - 1
            If Vi(k, 0) = 0 Then
                Lambda = Vr(k, 0)
                X = Er.ColVector(k)
            End If
        Next
        Dim total As Double = 0
        For k = 0 To GridRating.VisibleRowCount - 1
            total = total + X(k, 0)
        Next
        Dim maxValue As Decimal = 0
        For k = 0 To GridRating.VisibleRowCount - 1
            Priority(k) = X(k, 0) / total
            If Priority(k) > maxValue Then
                maxValue = Priority(k)
            End If
        Next
        For k = 0 To GridRating.VisibleRowCount - 1
            Priority(k) = Round(Priority(k) / maxValue, 3)
        Next
        For i = 0 To GridRating.VisibleRowCount - 1
            Dim drNode As DataRow
            For Each drNode In objdsAssessment.TblIntensity
                If drNode("RateID") = Id(i) Then
                    drNode("Priority") = Priority(i)
                End If
            Next
        Next
    End Sub

    Private Sub CheckChange()
        Dim objChanges As DFOdemo.dsAssessment = New DFOdemo.dsAssessment()
        objChanges = CType(objdsAssessment.GetChanges, DFOdemo.dsAssessment)
        If (Not (objChanges) Is Nothing) Then
            Dim sConfirm As String = String.Format("Do you want to save the change?")
```

```
        selResult = MessageBox.Show(sConfirm, "Comfirm Save!", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
        If selResult = DialogResult.Yes Then
            Try
                Me.UpdateDataSet()
            Catch eUpdate As System.Exception
                System.Windows.Forms.MessageBox.Show(eUpdate.Message)
            End Try
        Else
            objdsAssessment.RejectChanges()
        End If
    End If
End Sub

Private Sub ClearEdit()
    selScheme = 0
    GridIntensity.Visible = False
    TextRate.Text = "New Scheme"
    BtnOK.Enabled = True
    BtnCancel.Enabled = False
    BtnAssess.Visible = False
    BtnCal.Visible = False
    PanSelect.Enabled = True
    PanAssess.Visible = False
End Sub
#End Region
```

#Region "Event Handle"

```
    Private Sub DFOrate_Setup_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
        CheckChange()
        ComboRate.SelectedValue = selScheme
    End Sub

    Private Sub GridRating_CurrentCellChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles GridRating.CurrentCellChanged
        curRow = GridRating.CurrentCell.RowNumber
        curColumn = GridRating.CurrentCell.ColumnNumber
        If GridRating(preRow, preColumn) <> 0 Then
            Dim selValue As Decimal = 1 / GridRating(preRow, preColumn)
            objdsAssessment.TblRating.DefaultView(preColumn - 1).Item(preRow + 1 + 2) = Round(selvalue,
3)
        End If
        LVerbalNode.Text = GridRating.Item(curRow, 0)
        RVerbalNode.Text = ChildOfNode(curColumn - 1)
        If Not (GridRating(curRow, curColumn) = 0) Then
            Select Case GridRating(curRow, curColumn)
                Case Is = 1
                    TrackBar.Value = 8
                Case Is < 1
                    Dim selValue As Integer = Round(1 / GridRating(curRow, curColumn))
                    If selValue = 1 Then
                        TrackBar.Value = 8
                    Else
                        TrackBar.Value = 9 - selvalue
                    End If
                Case Is > 1
                    Dim selValue As Integer = Round(GridRating(curRow, curColumn))
                    If selValue = 1 Then
                        TrackBar.Value = 8
                    Else
                        TrackBar.Value = selvalue + 7
                    End If
            End Select
        Else
            TrackBar.Value = 8
        End If
        preRow = curRow
        preColumn = curColumn
    End Sub

    Private Sub TrackBar_MouseUp(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles TrackBar.MouseUp
        curRow = GridRating.CurrentCell.RowNumber
```

```
        curColumn = GridRating.CurrentCell.ColumnNumber
        Dim diaRow As Integer = curColumn - 1
        Dim diaColumn As Integer = curRow + 1
        Dim selValue As Decimal
        If curRow = curColumn - 1 Then
            GridRating(curRow, curColumn) = 1
            TrackBar.Value = 8
        Else
            Select Case TrackBar.Value
                Case Is = 8
                    GridRating(curRow, curColumn) = 1
                Case Is < 8
                    selValue = 1 / (9 - TrackBar.Value)
                    GridRating(curRow, curColumn) = Round(selValue, 3)
                    objdsAssessment.TblRating.DefaultView(diaRow).Item(diaColumn + 2) = 9 - TrackBar.Value
                Case Is > 8
                    selValue = 1 / (TrackBar.Value - 7)
                    GridRating(curRow, curColumn) = TrackBar.Value - 7
                    objdsAssessment.TblRating.DefaultView(diaRow).Item(diaColumn + 2) = Round(selValue,
3)
            End Select
        End If
    End Sub
#End Region
```

#Region "Button Click"

```
    Private Sub ButOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnOK.Click
        GridIntensity.Visible = True
        BtnOK.Enabled = False
        BtnCancel.Enabled = True
        BtnAssess.Visible = True
        PanSelect.Enabled = False
        ready = True
        If RadButNew.Checked = True Then
            Dim drScheme As DataRow
            For Each drScheme In objdsAssessment.TblScheme.Rows
                If drScheme.Item(1) = TextRate.Text Then
                    sConfirm = Format("The scheme is already existed!")
                    MessageBox.Show(sConfirm, "Confirm !", MessageBoxButtons.OK,
MessageBoxIcon.Information)
                    ready = False
                End If
            Next
            If ready Then
                drScheme = objdsAssessment.TblScheme.NewRow
                drScheme.Item(1) = TextRate.Text
                objdsAssessment.TblScheme.Rows.Add(drScheme)
                selScheme = drScheme.Item(0)
                ResetDatagrid(GridIntensity)
                SetIntensity()
            Else
                GridIntensity.Visible = False
                TextRate.Focus()
            End If
        Else
            selScheme = ComboRate.SelectedValue
            ResetDatagrid(GridIntensity)
            SetIntensity()
        End If
    End Sub

    Private Sub RadButNew_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles RadButNew.CheckedChanged
        ComboRate.Enabled = False
        TextRate.Visible = True
    End Sub

    Private Sub RadButOld_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles RadButOld.CheckedChanged
        ComboRate.Enabled = True
        TextRate.Visible = False
    End Sub
```

```
    Private Sub ButCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnCancel.Click
        CheckChange()
        ClearEdit()
    End Sub

    Private Sub BtnAssess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnAssess.Click
        If GridIntensity.VisibleRowCount < 2 Then
            sConfirm = Format("There is not enough rating scale to be assessed!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            PanAssess.Visible = False
        Else
            BtnCal.Visible = True
            GridIntensity.Enabled = False
            PanAssess.Visible = True
            LoadRating()
            ResetDatagrid(GridRating)
            SetRating()
            ready = False
        End If
    End Sub

    Private Sub BtnCal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnCal.Click
        CalPairwise()
    End Sub
#End Region
End Class
```

Module 3 Synthesis

This module finish the systhesis report including numerical, and graphical results of prioritization, which is designed as the step 3a in the flow chart (Figure 4-1).

```
Imports ChartDirector
Imports System.Math
Public Class DFOsynthesis
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim sConfirm, selResult, selName As String
    Dim selModelID, selNodeID, selParentID, selOption, selType, ModelL1, ModelL2, ModelL3, ModelL4,
currentTab As Integer
    Dim selNode, copyNode As TreeNode
    Dim selTable As DataTable
    Dim found As Boolean = True
    Private Sub DFOsynthesis_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        Me.Show()
        Dim frmNew As New DFOmodel_new()
        frmNew.ComboGoal.Visible = True
        frmNew.ComboGoal.Text = Nothing
        frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
        frmNew.ComboGoal.DisplayMember = "ModelName"
        frmNew.ComboGoal.ValueMember = "ModelType"
        frmNew.ShowDialog(Me)
        If frmNew.DialogResult = DialogResult.OK Then
            TreeView1.TopNode().Text = frmNew.ComboGoal.Text
            TreeView1.TopNode.ForeColor = Color.Blue
            selModelID = SearchModelId(TreeView1.TopNode, objdsAssessment.TblModel)
            LoadTreeView(selModelID, TreeView1.TopNode)
            LoadOptions()
        Else
            Me.Close()
        End If
    End Sub
```

'The following codes perform the event raised from the main menu of Module 3.1 including showing different report.

#Region "Menu Item"

```
    Private Sub MenuOpen_Click (See same Sub in Module 2)

    Private Sub MenuBar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuBar.Click
        Dim strExpr As String = Format("ModelID = " & selModelID)
        Dim OptionView As DataView = New DataView(objdsAssessment.TblOptions)
        OptionView.RowFilter = strExpr
        OptionView.Sort = "Priority"
        Dim selRow As Integer = OptionView.Count
        Dim data(selRow - 1) As Double
        Dim label(selRow - 1) As String
        Dim i As Integer
        For i = 0 To selRow - 1
            data(i) = OptionView(i).Item("Priority")
            label(i) = OptionView(i).Item("NodeName")
            If data(i) = 0 Then
                found = False
            End If
        Next
        If found Then
            TabSynthesis.SelectedIndex = 1
            Label1.Text = "Synthesis result with respect to " & TreeView1.TopNode.Text
            Label1.ForeColor = Color.DarkBlue
            createBarChart(BarViewer, data, label)
        Else
            TabSynthesis.SelectedIndex = 0
            sConfirm = Format("The assessment for this model has not been finished yet!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            found = True
        End If
    End Sub

    Private Sub MenuPie_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuPie.Click
        TabSynthesis.SelectedIndex = 2
    End Sub

    Private Sub MenuAbout_Click(See same Sub in Module 1)

    Private Sub MenuExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuExit.Click
        Me.Close()
    End Sub
#End Region
```

' The following code contain the sub procedures needed in the previous event handle and menu operation. These sub procedures perform the detailed step of most of the functions in this module.

#Region "Sub Procedure"

```
    Public Sub LoadDataSet (See same Sub in Module 1)

    Public Sub FillDataSet (See same Sub in Module 1)

    Public Sub UpdateDataSet (See same Sub in Module 1)

    Public Sub UpdateDataSource (See same Sub in Module 1)

    Public Sub LoadTreeView(ByVal selModelID As Integer, ByVal TopNode As TreeNode)
        Dim i, n As Integer
        Dim all As Boolean = True
        Dim parentrow As DataRow
        Dim ParentTable As DataTable
        ParentTable = objdsAssessment.Tables("TblNodeL1")
        For Each parentrow In ParentTable.Rows
```

```
            If parentrow.Item(2) = selModelID Then
                Dim parentnode As TreeNode
                parentnode = New TreeNode(parentrow.Item(1))
                parentnode.ForeColor = Color.Blue
                If parentrow.Item("Priority") > 0 Then
                    parentnode.ImageIndex = 12
                    parentnode.Text = parentnode.Text & "  [" & CStr(parentrow.Item("Priority")) & "]"
                Else
                    all = False
                End If
                TopNode.Nodes.Add(parentnode)
                Dim childrow As DataRow
                Dim childnode As TreeNode
                childnode = New TreeNode()
                For Each childrow In parentrow.GetChildRows("L1ToL2")
                    If childrow.Item("Priority") > 0 Then
                        childnode = parentnode.Nodes.Add(childrow.Item(1) & " [" &
CStr(childrow.Item("Priority")) & "]")
                        childnode.ImageIndex = 12
                    Else
                        childnode = parentnode.Nodes.Add(childrow.Item(1))
                        all = False
                    End If
                    childnode.ForeColor = Color.Blue
                Next childrow
            End If
        Next parentrow
        If all Then
            TreeView1.TopNode.ImageIndex = 12
        End If
        TreeView1.ExpandAll()
    End Sub

    Public Sub LoadOptions()
        LisOptions.Items.Clear()
        Dim all As Boolean = True
        Dim OptionsView As DataView = New DataView(objdsAssessment.TblOptions)
        Dim intCount As Integer
        Dim intCounter As Integer = 0
        Dim strExp As String = Format("ModelID = " & selModelID)
        OptionsView.RowFilter = strExp
        intCount = OptionsView.Count
        For intCounter = intCounter To intCount - 1
            If OptionsView.Item(intCounter).Item("Priority") = 0 Then
                all = False
                LisOptions.Items.Add(OptionsView.Item(intCounter).Item(1))
            Else
                Dim optionText As String
                optionText = OptionsView.Item(intCounter).Item(1) & " [" &
CStr(OptionsView.Item(intCounter).Item("Priority")) & "]"
                LisOptions.Items.Add(optionText)
            End If
        Next
        If all Then
            LisOptions.ForeColor = Color.Blue
        Else
            LisOptions.ForeColor = Color.Red
        End If
    End Sub

    Public Function SearchId (See same Sub in Module 1)

    Public Function SearchModelId (See same Sub in Module 1)

    Private Sub ClearEditModel (See same Sub in Module 2)

    Public Sub createBarChart(ByVal viewer As WinChartViewer, ByVal data As Array, ByVal labels() As
String)
        Dim c As XYChart = New XYChart(1036, 900)
        c.addTitle(" ", "arialbi.ttf", 12, Chart.CColor(Color.DarkBlue))
        c.setPlotArea(400, 30, 400, 200, Chart.Transparent, Chart.Transparent, Chart.Transparent,
Chart.Transparent, Chart.Transparent)
        Dim layer As BarLayer = c.addBarLayer(data, c.gradientColor(300, 0, 800, 0,
Chart.CColor(Color.RoyalBlue), Chart.CColor(Color.White)))
        layer.setBorderColor(Chart.Transparent)
        c.swapXY(True)
        layer.setBarGap(0.2)
```

```
        layer.setAggregateLabelFormat("{value}")
        layer.setAggregateLabelStyle("timesbi.ttf", 11, Chart.CColor(Color.RoyalBlue))
        Dim textbox As ChartDirector.TextBox = c.xAxis().setLabels(labels)
        textbox.setFontStyle("arialbi.ttf")
        textbox.setFontSize(12)
        c.xAxis().setColors(Chart.Transparent, Chart.CColor(Color.DarkBlue))
        c.yAxis().setColors(Chart.Transparent, Chart.Transparent)
        viewer.Image = c.makeImage()
        viewer.ImageMap = c.getHTMLImageMap("clickable", "", _
            "title='{xLabel}: {value} '")
    End Sub

    Public Sub createPieChart(ByVal viewer As WinChartViewer, ByVal data As Array, ByVal labels() As
String)
        Dim c As PieChart = New PieChart(1036, 900)
        c.setPieSize(500, 300, 150)
        c.addTitle2(Chart.TopCenter, "Synthesis with respect to " & TreeView1.TopNode.Text, "arialbi.ttf",
14, Chart.CColor(Color.DarkBlue))
        c.set3D()
        c.setLabelFormat("{label} ${value}K<*br*>({percent}%)")
        c.setLabelStyle("timesbi.ttf", 11, Chart.CColor(Color.Blue))
        c.setData(data, labels)
        c.setExplode(0)
        viewer.Image = c.makeImage()
        viewer.ImageMap = c.getHTMLImageMap("clickable", "", _
            "title='{label}: US${value}K ({percent}%)'")
    End Sub
#End Region
```

' The following code handle the event raised by certain operation other than selecting the menu such as tab switch, double click the buttons.

#Region "Event Handle"

```
    Private Sub TabSynthesis_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabSynthesis.SelectedIndexChanged
        Select Case TabSynthesis.SelectedIndex
            Case 1
                Dim strExpr As String = Format("ModelID = " & selModelID)
                Dim OptionView As DataView = New DataView(objdsAssessment.TblOptions)
                OptionView.RowFilter = strExpr
                OptionView.Sort = "Priority"
                Dim selRow As Integer = OptionView.Count
                Dim data(selRow - 1) As Double
                Dim label(selRow - 1) As String
                Dim i As Integer
                For i = 0 To selRow - 1
                    data(i) = OptionView(i).Item("Priority")
                    label(i) = OptionView(i).Item("NodeName")
                    If data(i) = 0 Then
                        found = False
                    End If
                Next
                If found Then
                    TabSynthesis.SelectedIndex = 1
                    Label1.Text = "Synthesis result with respect to " & TreeView1.TopNode.Text
                    Label1.ForeColor = Color.DarkBlue
                    createBarChart(BarViewer, data, label)
                Else
                    TabSynthesis.SelectedIndex = 0
                    sConfirm = Format("The assessment for this model has not been finished yet!")
                    MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                    found = True
                End If
            Case 2
                PieViewer.Visible = False
        End Select
    End Sub

    Private Sub BtnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnShow.Click
        Dim strExpr As String = Format("ModelID = " & selModelID)
        Dim foundRow() As DataRow
        foundRow = objdsAssessment.TblModel.Select(strExpr)
        foundRow(0).Item("Allocation") = CInt(TexValue.Text)
```

```
        Dim OptionView As DataView = New DataView(objdsAssessment.TblOptions)
        OptionView.RowFilter = strExpr
        OptionView.Sort = "Priority"
        Dim selRow As Integer = OptionView.Count
        Dim data(selRow - 1) As Double
        Dim label(selRow - 1) As String
        Dim OptionID(selRow - 1) As Integer
        Dim i As Integer
        For i = 0 To selRow - 1
            data(i) = OptionView(i).Item("Priority") * CInt(TexValue.Text)
            label(i) = OptionView(i).Item("NodeName")
            OptionID(i) = OptionView(i).Item("OptionID")
            If data(i) = 0 Then
                found = False
            End If
        Next
        If found Then
            For i = 0 To selRow - 1
                Dim drModel As DataRow
                For Each drModel In objdsAssessment.TblModel
                    If drModel.Item("OptionID") = OptionID(i) Then
                        drModel.Item("Allocation") = Round(data(i), 1)
                    End If
                Next
            Next
            PieViewer.Visible = True
            createPieChart(PieViewer, data, label)
        Else
            TabSynthesis.SelectedIndex = 0
            sConfirm = Format("The assessment for this model has not been finished yet!")
            MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            found = True
        End If
    End Sub

    Private Sub BtnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnClear.Click
        TexValue.Text = " "
        PieViewer.Visible = False
    End Sub

    Private Sub BtnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnSave.Click
        Try
            Me.UpdateDataSet()
        Catch eUpdate As System.Exception
            System.Windows.Forms.MessageBox.Show(eUpdate.Message)
        End Try
    End Sub

    Private Sub ToolBar2_ButtonClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar2.ButtonClick
        Select Case ToolBar2.Buttons.IndexOf(e.Button)
            Case 0
                Dim frmNew As New DFOmodel_new()
                frmNew.ComboGoal.Visible = True
                frmNew.ComboGoal.Text = Nothing
                frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
                frmNew.ComboGoal.DisplayMember = "ModelName"
                frmNew.ComboGoal.ValueMember = "ModelType"
                frmNew.ShowDialog(Me)
                If frmNew.DialogResult = DialogResult.OK Then
                    ClearEditModel()
                    TreeView1.TopNode().Text = frmNew.ComboGoal.Text
                    TreeView1.TopNode.ForeColor = Color.Blue
                    selModelID = SearchModelId(TreeView1.TopNode, objdsAssessment.TblModel)
                    LoadTreeView(selModelID, TreeView1.TopNode)
                    LoadOptions()
                End If
        End Select
    End Sub
#End Region
End Class
```

Module 4 Risk Analysis

Module 4 is the last module of system. This module performs risk analysis at the end of science prioritization. According to the flow chart in Figure 4-1, it implements the step 3b and 3c. The functions of this module include top-down dynamic, bottom-up dynamic analysis and utility analysis.

```
Imports ChartDirector
Imports System.Math
Imports BluebitMatrix30
Public Class DFOsensitivity
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    'Please refer to the source code in VB.NET
#End Region

    Dim sConfirm, selResult, selName, labelOption(13) As String
    Dim selModelID, selObjRow, selOptRow, index, curRow, curColumn As Integer
    Dim ChildOfNode(20) As String
    Dim Percent(20) As Decimal
    Dim Budget As Decimal = 0
    Dim found As Boolean = True
    Dim data(10), dataoption(10), originalData(10) As Double
    Dim Ready As Boolean = True
    Dim temp As New Matrix()
    Dim NodeData As New Matrix()
    Dim OptionData As New Matrix()
    Private Sub DFOsensitivity_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Try
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        Me.Show()
        Dim frmNew As New DFOmodel_new()
        frmNew.ComboGoal.Visible = True
        frmNew.ComboGoal.Text = Nothing
        frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
        frmNew.ComboGoal.DisplayMember = "ModelName"
        frmNew.ComboGoal.ValueMember = "ModelType"
        frmNew.ShowDialog(Me)
        If frmNew.DialogResult = DialogResult.OK Then
            TreeView1.TopNode().Text = frmNew.ComboGoal.Text
            TreeView1.TopNode.ForeColor = Color.Blue
            selModelID = SearchModelId(TreeView1.TopNode, objdsAssessment.TblModel)
            LoadTreeView(selModelID, TreeView1.TopNode)
            LoadOptions()
        Else
            Me.Close()
        End If
    End Sub
```

'The following codes perform the event raised from the main menu of Module 4 including different type of risk analysis.

#Region "Menu Item"

```
    Private Sub MenuOpen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuOpen.Click
        Dim frmNew As New DFOmodel_new()
        frmNew.ComboGoal.Visible = True
        frmNew.ComboGoal.Text = Nothing
        frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
        frmNew.ComboGoal.DisplayMember = "ModelName"
        frmNew.ComboGoal.ValueMember = "ModelType"
        frmNew.ShowDialog(Me)
        If frmNew.DialogResult = DialogResult.OK Then
            TreeView1.TopNode().Text = frmNew.ComboGoal.Text
            selModelID = SearchModelId(TreeView1.TopNode, objdsAssessment.TblModel)
            LoadTreeView(selModelID, TreeView1.TopNode)
```

```
            LoadOptions()
            Me.TabSensitivity.SelectedIndex = 0
        End If
    End Sub

    Private Sub MenuAbout_Click (See same Sub in Module 1)

    Private Sub MenuExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuExit.Click
        Me.Close()
    End Sub

    Private Sub MenuTop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuTop.Click
        TabSensitivity.SelectedIndex = 1
    End Sub

    Private Sub MenuBottom_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuBottom.Click
        TabSensitivity.SelectedIndex = 2
    End Sub

    Private Sub MenuUtility_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MenuUtility.Click
        TabSensitivity.SelectedIndex = 3
    End Sub
#End Region
```

'The following code contain the sub procedures needed in the previous event handle and menu operation. These sub procedures perform the detailed step of most of the functions in this module.

#Region "Sub Procedure"

```
    Public Sub LoadDataSet (See same Sub in Module 1)

    Public Sub FillDataSet (See same Sub in Module 1)

    Public Sub LoadTreeView (See same Sub in Module 1)

    Public Sub LoadOptions (See same Sub in Module 1)

    Public Function SearchId (See same Sub in Module 1)

    Public Function SearchModelId (See same Sub in Module 1)

    Private Function SearchTable (See same Sub in Module 1)

    Private Sub ClearEditModel (See same Sub in Module 2)

    Public Sub MenuCalDatagrid()
        Dim strExpr As String
        strExpr = Format("ModelID = " & selModelID)
        Dim foundRows() As DataRow
        foundRows = objdsAssessment.TblNodeL1.Select(strExpr)
        selObjRow = foundRows.GetUpperBound(0)
        Dim tempTotal As New Matrix()
        temp.Size(selObjRow + 1, 1)
        NodeData.Size(selObjRow + 1, 6)
        OptionData.Size(selObjRow + 1, 6)
        tempTotal.Size(selObjRow + 1, 6)
        Dim temp0 As New Matrix()
        temp0.Size(1, selObjRow + 1)
        foundRows = objdsAssessment.TblDatagrid.Select(strExpr)
        index = foundRows.GetUpperBound(0)
        Dim priority(index + 1) As Double
        Dim selOptions(index + 1) As Integer
        Dim i, j As Integer
        For i = 0 To index
            selOptions(i) = foundRows(i).Item(0)
            curRow = i
            curColumn = 0
            OptionData.Clear()
            PriorityL1(TreeView1.TopNode.FirstNode, 0)
            tempTotal = tempTotal.Plus(OptionData)
```

```
        Next
        For i = 0 To index
            selOptions(i) = foundRows(i).Item(0)
            curRow = i
            curColumn = 0
            temp.Clear()
            NodeData.Clear()
            OptionData.Clear()
            priority(i) = 0
            PriorityL1(TreeView1.TopNode.FirstNode, 0)
            Dim k As Integer
            For j = 0 To selObjRow
                For k = 0 To 5
                    If tempTotal(j, k) <> 0 Then
                        OptionData(j, k) = OptionData(j, k) / tempTotal(j, k)
                    End If
                Next
            Next
            Dim A As New Matrix()
            A = OptionData.Transpose()
            A = NodeData.Times(A)
            Dim x, y As Integer
            For x = 0 To selObjRow
                For y = 0 To selObjRow
                    If x = y Then
                        temp0(0, x) = A(x, x)
                    End If
                Next
            Next
            A = temp0.Times(temp)
            priority(i) = A(0, 0)
            dataoption(i) = Round(priority(i), 3) * 100
        Next
    End Sub

    Private Sub PriorityL1(ByVal sNode As TreeNode, ByVal k As Integer)
        Dim sNodeID As Integer = SearchId(sNode, selModelID, objdsAssessment.TblNodeL1)
        If TabSensitivity.SelectedIndex = 1 Then
            temp(k, 0) = data(k) / 100
        End If
        If TabSensitivity.SelectedIndex = 2 Then
            Dim drNode As DataRow
            For Each drNode In objdsAssessment.TblNodeL1.Rows
                If drNode("NodeID") = sNodeID Then
                    temp(k, 0) = 0
                    temp(k, 0) = drNode("Priority")
                End If
            Next
        End If
        If Not (sNode.FirstNode Is Nothing) Then
            PriorityL2(sNode.FirstNode, sNodeID, k, 0)
        End If
        If Not (sNode.NextNode Is Nothing) Then
            curColumn += 1
            k += 1
            PriorityL1(sNode.NextNode, k)
        End If
    End Sub

    Private Sub PriorityL2(ByVal sNode As TreeNode, ByVal sParent As Integer, ByVal row As Integer, ByVal
column As Integer)
        Dim sNodeID As Integer = SearchId(sNode, sParent, objdsAssessment.TblNodeL2)
        Dim drNode As DataRow
        For Each drNode In objdsAssessment.TblNodeL2.Rows
            If drNode("NodeID") = sNodeID Then
                NodeData(row, column) = 0
                NodeData(row, column) = drNode("Priority")
                If TabSensitivity.SelectedIndex = 1 Then
                    OptionData(row, column) = DatagridOption(curRow, curColumn + 5)
                End If
                If TabSensitivity.SelectedIndex = 2 Then
                    OptionData(row, column) = GridOption(curRow, curColumn + 5)
                End If
            End If
        Next
        If Not (sNode.NextNode Is Nothing) Then
            curColumn += 1
```

```
        column += 1
        PriorityL2(sNode.NextNode, sParent, row, column)
    End If
End Sub
```

' Post the slider according to options' priorities in top-down

```
Private Sub postOptionWeight()
    Dim count As Integer = 1
    If Not (count > selOptRow) Then
        RL0.Text = labelOption(0)
        RP0.Visible = True
        RP00.Visible = True
        RP0.PortionValue = dataoption(0)
        RP00.PortionValue = originalData(0)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL1.Text = labelOption(1)
        RP1.Visible = True
        RP10.Visible = True
        RP1.PortionValue = dataoption(1)
        RP10.PortionValue = originalData(1)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL2.Text = labelOption(2)
        RP2.Visible = True
        RP20.Visible = True
        RP2.PortionValue = dataoption(2)
        RP20.PortionValue = originalData(2)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL3.Text = labelOption(3)
        RP3.Visible = True
        RP30.Visible = True
        RP3.PortionValue = dataoption(3)
        RP30.PortionValue = originalData(3)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL4.Text = labelOption(4)
        RP4.Visible = True
        RP40.Visible = True
        RP4.PortionValue = dataoption(4)
        RP40.PortionValue = originalData(4)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL5.Text = labelOption(5)
        RP5.Visible = True
        RP50.Visible = True
        RP5.PortionValue = dataoption(5)
        RP50.PortionValue = originalData(5)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL6.Text = labelOption(6)
        RP6.Visible = True
        RP60.Visible = True
        RP6.PortionValue = dataoption(6)
        RP60.PortionValue = originalData(6)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL7.Text = labelOption(7)
        RP7.Visible = True
        Rp70.Visible = True
        RP7.PortionValue = dataoption(7)
        Rp70.PortionValue = originalData(7)
        count += 1
    End If
    If Not (count > selOptRow) Then
        RL8.Text = labelOption(8)
        RP8.Visible = True
        RP80.Visible = True
```

```
            RP8.PortionValue = dataoption(8)
            RP80.PortionValue = originalData(8)
            count += 1
        End If
        If Not (count > selOptRow) Then
            RL9.Text = labelOption(9)
            RP9.Visible = True
            RP90.Visible = True
            RP9.PortionValue = dataoption(9)
            RP90.PortionValue = originalData(9)
            count += 1
        End If
    End Sub
```

' Post the slider according to options' priorities in bottom-up

```
    Private Sub OptionWeight()
        Dim count As Integer = 1
        If Not (count > selOptRow) Then
            LblOption1.Text = labelOption(0)
            Percent0.Visible = True
            Percent00.Visible = True
            Percent0.PortionValue = dataoption(0)
            Percent00.PortionValue = originalData(0)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption2.Text = labelOption(1)
            Percent1.Visible = True
            Percent10.Visible = True
            Percent1.PortionValue = dataoption(1)
            Percent10.PortionValue = originalData(1)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption3.Text = labelOption(2)
            Percent2.Visible = True
            Percent20.Visible = True
            Percent2.PortionValue = dataoption(2)
            Percent20.PortionValue = originalData(2)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption4.Text = labelOption(3)
            Percent3.Visible = True
            Percent30.Visible = True
            Percent3.PortionValue = dataoption(3)
            Percent30.PortionValue = originalData(3)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption5.Text = labelOption(4)
            Percent4.Visible = True
            Percent40.Visible = True
            Percent4.PortionValue = dataoption(4)
            Percent40.PortionValue = originalData(4)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption6.Text = labelOption(5)
            Percent5.Visible = True
            Percent50.Visible = True
            Percent5.PortionValue = dataoption(5)
            Percent50.PortionValue = originalData(5)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption7.Text = labelOption(6)
            Percent6.Visible = True
            Percent60.Visible = True
            Percent6.PortionValue = dataoption(6)
            Percent60.PortionValue = originalData(6)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption8.Text = labelOption(7)
            Percent7.Visible = True
```

```
            Percent70.Visible = True
            Percent7.PortionValue = dataoption(7)
            Percent70.PortionValue = originalData(7)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption9.Text = labelOption(8)
            Percent8.Visible = True
            Percent80.Visible = True
            Percent8.PortionValue = dataoption(8)
            Percent80.PortionValue = originalData(8)
            count += 1
        End If
        If Not (count > selOptRow) Then
            LblOption10.Text = labelOption(9)
            Percent9.Visible = True
            Percent90.Visible = True
            Percent9.PortionValue = dataoption(9)
            Percent90.PortionValue = originalData(9)
            count += 1
        End If
    End Sub
```

' Post the datagrid value for selected alternative in bottom-up

```
    Private Sub SetListBox(ByVal sOption As Integer)
        LstBoxGrid.Items.Clear()
        Dim i As Integer
        For i = 0 To selOptRow - 1
            If GridOption(i, 0) = sOption Then
                For index = 5 To 24
                    If GridOption(i, index) <> 0 Then
                        LstBoxGrid.Items.Add(ChildOfNode(index - 5) & "     [ " & CStr(GridOption(i, index))
& " ]")
                    End If
                Next
            End If
        Next
    End Sub
    Private Sub GetChild(ByVal sNode As TreeNode)
        Dim aNode As TreeNode
        For Each aNode In sNode.Nodes
            ChildOfNode(index) = sNode.Text & " -- " & aNode.Text
            index += 1
        Next
        If Not (sNode.NextNode Is Nothing) Then
            GetChild(sNode.NextNode)
        End If
    End Sub
```

' Create utility chart for single alternative

```
    Public Sub createChart(ByVal viewer As WinChartViewer, ByVal grid As DataGrid)
        Dim row As Integer = grid.CurrentRowIndex
        LblUtility.Text = "Utility Curve - " & grid(row, 0)
        Dim data0() As Double = {0, 0.2, 0.5, 0.8, 1}
        Dim data1() As Double = {0, grid(row, 1), grid(row, 2), grid(row, 3), grid(row, 4)}
        Dim labels() As String = {"0", "2", "4", "6", "8", "10"}
        Dim labelY() As String = {"0%", "20%", "40%", "60%", "80%", "100%", " "}
        Dim c As XYChart = New XYChart(500, 800, &HFFFFFF, &H0, 1)
        c.setPlotArea(50, 25, 420, 205, &HFFF0C0, -1, &HA08040, &HA08040)
        c.addLegend(200, 0, False, "", 8).setBackground(Chart.Transparent)
        c.yAxis().setTitle("Utility")
        c.xAxis().setTitle("<*block,valign=absmiddle*>  Program Allocation (Budget x Program Proportion)
")
        c.xAxis().setWidth(2)
        c.yAxis().setWidth(2)
        Dim layer As SplineLayer = c.addSplineLayer()
        layer.setLineWidth(2)
        layer.addDataSet(data0, &HC0, "Thresholds").setDataSymbol(Chart.DiamondSymbol, 9, &HFFFF00)
        layer.setXData(data1)
        Dim x As ChartDirector.TextBox = c.addText(10, 25, " ")
        x.setPos(20, 20)
        x.setSize(480, 36)
        x.setBackground(Chart.CColor(Color.White), -1)
        x.setText(" ")
        viewer.Image = c.makeImage()
    End Sub
```

' Create total utility chart

```
    Public Sub createTotal(ByVal viewer As WinChartViewer, ByVal grid As DataGrid)
        Dim row As Integer
        Dim data() As Double = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
        Dim data01() As Double = {0}
        Dim data0() As Double = {0}
        Dim data1() As Double = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
        Dim TotalBudget As Decimal = CInt(TxtBudget.Text)
        Dim discount As Decimal = 0.5
        Dim U(grid.VisibleRowCount) As Decimal
        data(0) = 0
        data0(0) = 0
        data1(0) = 0
        Dim i As Integer
        For i = 1 To 11
            data(i) = 0
            U.Initialize()
            Dim intercept, slope As Decimal
            data1(i) = TotalBudget * discount
            For row = 0 To grid.VisibleRowCount - 1
                Dim Money As Decimal = Percent(row) * data1(i)
                If Money < grid(row, 1) Then
                    slope = (0 - 0.2) / (0 - grid(row, 1))
                    intercept = 0.2 - grid(row, 1) * ((0 - 0.2) / (0 - grid(row, 1)))
                    U(row) = slope * Money + intercept
                Else
                    If Money < grid(row, 2) Then
                        slope = (0.2 - 0.5) / (grid(row, 1) - grid(row, 2))
                        intercept = 0.5 - grid(row, 2) * (0.2 - 0.5) / (grid(row, 1) - grid(row, 2))
                        U(row) = slope * Money + intercept
                    Else
                        If Money < grid(row, 3) Then
                            slope = (0.5 - 0.8) / (grid(row, 2) - grid(row, 3))
                            intercept = 0.8 - grid(row, 3) * (0.5 - 0.8) / (grid(row, 2) - grid(row, 3))
                            U(row) = slope * Money + intercept
                        Else
                            slope = (0.8 - 1) / (grid(row, 3) - grid(row, 4))
                            intercept = 1 - grid(row, 4) * (0.8 - 1) / (grid(row, 3) - grid(row, 4))
                            U(row) = slope * Money + intercept
                        End If
                    End If
                End If
                If U(row) > 1 Then
                    U(row) = 1
                End If
                data(i) = data(i) + U(row)
            Next
            discount += 0.1
        Next
        data0(0) = data(6)
        data01(0) = data1(6)
        Dim labels() As String = {" -50%", " -40%", " -30%", " -20%", " -10%", "Current", "+10%", "+20%",
"+30%", "+40%", "+50%"}
        Dim colors() As Integer = {&HC3C3E6, &HC3C3E6, &HC3C3E6, &HC3C3E6, &HC3C3E6, &HC3C3E6, &H333366,
&HC3C3E6, &HC3C3E6, &HC3C3E6, &HC3C3E6, &HC3C3E6}
        Dim c As XYChart = New XYChart(800, 840)
        c.setBackground(c.gradientColor(Chart.goldGradient), -1, 2)
        c.addTitle("<*block,valign=absmiddle*>   Total   Utility   Assessment   ",   "arialbd.ttf",
10).setBackground(c.gradientColor(Chart.blueMetalGradient), -1, 1)
        c.setPlotArea(70, 50, 660, 240)
        Dim layer As SplineLayer = c.addSplineLayer()
        layer.setLineWidth(2)
        layer.addDataSet(data, &HC0)
        layer.setXData(data1)
        Dim layer1 As BarLayer = c.addBarLayer3(data0, colors)
        layer1.setBorderColor(-1, 1)
        layer1.setXData(data01)
        layer1.setBarWidth(10)
        c.layout()
        c.addText(layer1.getXCoor(data01(0)) + 5, layer1.getYCoor(0.5) - 10, "Current Budget: " &
TxtBudget.Text, "timesbi.ttf", 11, &H663300)
        c.yAxis().setTitle("Total Utility")
        c.xAxis().setTitle("<*block,valign=absmiddle*> Total Budget ")
        Dim x As ChartDirector.TextBox = c.addText(10, 25, " ")
        x.setPos(10, 20)
```

```
        x.setSize(780, 40)
        x.setBackground(c.gradientColor(Chart.goldGradient), -1)
        x.setText(" ")
        viewer.Image = c.makeImage()
    End Sub
#End Region
```

'The following codes handle the event raised by certain operation other than selecting the menu such as tab switch, slider's movement.

#Region "Event Handle"

```
    Private Sub TabSensitivity_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TabSensitivity.SelectedIndexChanged
        Dim i As Integer
        Select Case TabSensitivity.SelectedIndex
            Case 1
                found = True
                Dim strExpr As String = Format("ModelID = " & selModelID)
                Dim ObjView As DataView = New DataView(objdsAssessment.TblNodeL1)
                ObjView.RowFilter = strExpr
                selObjRow = ObjView.Count
                data.Clear(data, 0, 10)
                Dim label(selObjRow - 1) As String
                For i = 0 To selObjRow - 1
                    data(i) = ObjView(i).Item("Priority") * 100
                    label(i) = ObjView(i).Item("NodeName")
                    If data(i) = 0 Then
                        found = False
                    End If
                Next
                If found Then
                    TabSensitivity.SelectedIndex = 1
                    Select Case selObjRow
                        Case 1
                            LL1.Text = label(0)
                            LP1.Visible = True
                            LP1.PortionValue = data(0)
                        Case 2
                            LL1.Text = label(0)
                            LP1.Visible = True
                            LP1.PortionValue = data(0)
                            LL2.Text = label(1)
                            LP2.Visible = True
                            LP2.PortionValue = data(1)
                        Case 3
                            LL1.Text = label(0)
                            LP1.Visible = True
                            LP1.PortionValue = data(0)
                            LL2.Text = label(1)
                            LP2.Visible = True
                            LP2.PortionValue = data(1)
                            LL3.Text = label(2)
                            LP3.Visible = True
                            LP3.PortionValue = data(2)
                        Case 4
                            LL1.Text = label(0)
                            LP1.Visible = True
                            LP1.PortionValue = data(0)
                            LL2.Text = label(1)
                            LP2.Visible = True
                            LP2.PortionValue = data(1)
                            LL3.Text = label(2)
                            LP3.Visible = True
                            LP3.PortionValue = data(2)
                            LL4.Text = label(3)
                            LP4.Visible = True
                            LP4.PortionValue = data(3)
                    End Select
                    strExpr = Format("ModelID = " & selModelID)
                    Dim OptionView As DataView = New DataView(objdsAssessment.TblOptions)
                    OptionView.RowFilter = strExpr
                    selOptRow = OptionView.Count
                    dataoption.Initialize()
                    originalData.Initialize()
```

```
                labelOption.Initialize()
                For i = 0 To selOptRow - 1
                    dataoption(i) = OptionView(i).Item("Priority") * 100
                    originalData(i) = dataoption(i)
                    labelOption(i) = OptionView(i).Item("NodeName")
                    If dataoption(i) = 0 Then
                        found = False
                    End If
                Next
                If found Then
                    TabSensitivity.SelectedIndex = 1
                    postOptionWeight()
                    With objdsAssessment.TblDatagrid
                        .DefaultView.RowFilter = "ModelID =" & selModelID
                        .DefaultView.AllowNew = False
                        DatagridOption.DataSource = .DefaultView
                    End With
                Else
                    TabSensitivity.SelectedIndex = 0
                    sConfirm = Format("The assessment for this model has not been finished yet!")
                    MessageBox.Show(sConfirm,        "Warning        !",        MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                    found = True
                End If
            Else
                TabSensitivity.SelectedIndex = 0
                sConfirm = Format("The assessment for this model has not been finished yet!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                found = True
            End If
        Case 2
            found = True
            With objdsAssessment.TblDatagrid
                .DefaultView.RowFilter = "ModelID =" & selModelID
                .DefaultView.AllowNew = False
                GridOption.DataSource = .DefaultView
            End With
            Dim strExpr = Format("ModelID = " & selModelID)
            Dim OptionView As DataView = New DataView(objdsAssessment.TblOptions)
            OptionView.RowFilter = strExpr
            selOptRow = OptionView.Count
            Ready = False
            ComboOptions.DataSource = OptionView
            ComboOptions.DisplayMember = "NodeName"
            ComboOptions.ValueMember = "OptionID"
            Ready = True
            dataoption.Initialize()
            originalData.Initialize()
            labelOption.Initialize()
            For i = 0 To selOptRow - 1
                dataoption(i) = OptionView(i).Item("Priority") * 100
                originalData(i) = dataoption(i)
                labelOption(i) = OptionView(i).Item("NodeName")
                If dataoption(i) = 0 Then
                    found = False
                End If
            Next
            If found Then
                TabSensitivity.SelectedIndex = 2
                OptionWeight()
                For index = 0 To 19
                    ChildOfNode(index) = " "
                Next
                index = 0
                GetChild(TreeView1.TopNode.FirstNode)
                SetListBox(ComboOptions.SelectedValue)
            Else
                TabSensitivity.SelectedIndex = 0
                sConfirm = Format("The assessment for this model has not been finished yet!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                found = True
            End If
        Case 3
            found = True
            Dim drModel As DataRow
            For Each drModel In objdsAssessment.TblModel
                If drModel.Item("ModelID") = selModelID Then
```

```
                Budget = drModel.Item("Allocation")
                If Budget = 0 Then
                    found = False
                End If
            End If
        Next
        TxtBudget.Text = CStr(Budget)
        With objdsAssessment.TblOptions
            .DefaultView.RowFilter = "ModelID =" & selModelID
            .DefaultView.Sort = "OptionID"
        End With
        Dim selOption As Integer = objdsAssessment.TblOptions.DefaultView.Count
        Dim midThreshold(selOption) As Decimal
        Percent.Initialize()
        For i = 0 To selOption - 1
            Percent(i) = objdsAssessment.TblOptions.DefaultView.Item(i).Item("Priority")
            midThreshold(i) = Percent(i) * Budget
            If midThreshold(i) = 0 Then
                found = False
            End If
        Next
        If found Then
            Dim Threshold As New DataTable()
            Dim myDataColumn As DataColumn
            Dim myDataRow As DataRow
            myDataColumn = New DataColumn()
            With myDataColumn
                .DataType = System.Type.GetType("System.String")
                .ColumnName = "Alternative"
                .ReadOnly = True
                .Unique = True
            End With
            Threshold.Columns.Add(myDataColumn)
            myDataColumn = New DataColumn()
            With myDataColumn
                .DataType = System.Type.GetType("System.Decimal")
                myDataColumn.ColumnName = "Low Threshold"
                myDataColumn.ReadOnly = False
                myDataColumn.Unique = False
            End With
            Threshold.Columns.Add(myDataColumn)
            myDataColumn = New DataColumn()
            With myDataColumn
                .DataType = System.Type.GetType("System.Decimal")
                .ColumnName = "Mid Threshold"
                .ReadOnly = False
                .Unique = False
            End With
            Threshold.Columns.Add(myDataColumn)
            myDataColumn = New DataColumn()
            With myDataColumn
                .DataType = System.Type.GetType("System.Decimal")
                .ColumnName = "High Threshold"
                .ReadOnly = False
                .Unique = False
            End With
            Threshold.Columns.Add(myDataColumn)
            myDataColumn = New DataColumn()
            With myDataColumn
                .DataType = System.Type.GetType("System.Decimal")
                .ColumnName = "Maximum"
                .ReadOnly = False
                .Unique = False
            End With
            Threshold.Columns.Add(myDataColumn)
            For i = 0 To selOption - 1
                myDataRow = Threshold.NewRow
                myDataRow.Item(0) =
objdsAssessment.TblOptions.DefaultView.Item(i).Item("NodeName")
                myDataRow.Item(1) = Round(midThreshold(i) * 10 / 7 * 0.5)
                myDataRow.Item(2) = Round(midThreshold(i))
                myDataRow.Item(3) = Round(midThreshold(i) * 10 / 7 * 0.8)
                myDataRow.Item(4) = Round(midThreshold(i) * 10 / 7)
                Threshold.Rows.Add(myDataRow)
            Next
            With Threshold
                .DefaultView.AllowNew = False
```

```
                End With
                With GridThreshold
                    .DataSource = Threshold.DefaultView
                End With
                createChart(UtilityForm, GridThreshold)
                createTotal(UtilityTotal, GridThreshold)
            Else
                TabSensitivity.SelectedIndex = 0
                sConfirm = Format("The assessment for this model has not been finished yet!")
                MessageBox.Show(sConfirm, "Warning !", MessageBoxButtons.OK, MessageBoxIcon.Warning)
                found = True
            End If
        End Select
    End Sub

    Private Sub ToolBar_ButtonClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar.ButtonClick
        Select Case ToolBar.Buttons.IndexOf(e.Button)
            Case 0
                Dim frmNew As New DFOmodel_new()
                frmNew.ComboGoal.Visible = True
                frmNew.ComboGoal.Text = Nothing
                frmNew.ComboGoal.DataSource = objdsAssessment.TblModel
                frmNew.ComboGoal.DisplayMember = "ModelName"
                frmNew.ComboGoal.ValueMember = "ModelType"
                frmNew.ShowDialog(Me)
                If frmNew.DialogResult = DialogResult.OK Then
                    ClearEditModel()
                    TreeView1.TopNode().Text = frmNew.ComboGoal.Text
                    TreeView1.TopNode.ForeColor = Color.Blue
                    selModelID = SearchModelId(TreeView1.TopNode, objdsAssessment.TblModel)
                    LoadTreeView(selModelID, TreeView1.TopNode)
                    LoadOptions()
                    Me.TabSensitivity.SelectedIndex = 0
                End If
        End Select
    End Sub
```

' Change the slider value for objective 1

```
    Private Sub LP1_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LP1.Change
        Dim oldTotal As Double = 100 - data(0)
        Dim newTotal As Double = 100 - LP1.PortionValue
        data(0) = LP1.PortionValue
        LP2.PortionValue = data(1) / oldTotal * newTotal
        data(1) = LP2.PortionValue
        LP3.PortionValue = data(2) / oldTotal * newTotal
        data(2) = LP3.PortionValue
        MenuCalDatagrid()
        postOptionWeight()
    End Sub
```

' Change the slider value for objective 2

```
    Private Sub LP2_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LP2.Change
        Dim oldTotal As Double = 100 - data(1)
        Dim newTotal As Double = 100 - LP2.PortionValue
        data(1) = LP2.PortionValue
        LP1.PortionValue = data(0) / oldTotal * newTotal
        data(0) = LP1.PortionValue
        LP3.PortionValue = data(2) / oldTotal * newTotal
        data(2) = LP3.PortionValue
        MenuCalDatagrid()
        postOptionWeight()
    End Sub
```

' Change the slider value for objective 3

```
    Private Sub LP3_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LP3.Change
        Dim oldTotal As Double = 100 - data(2)
        Dim newTotal As Double = 100 - LP3.PortionValue
        data(2) = LP3.PortionValue
        LP1.PortionValue = data(0) / oldTotal * newTotal
        data(0) = LP1.PortionValue
        LP2.PortionValue = data(1) / oldTotal * newTotal
        data(1) = LP2.PortionValue
        MenuCalDatagrid()
        postOptionWeight()
    End Sub
```

' Select alternative in bottom-up

```
   Private Sub ComboOptions_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboOptions.SelectedIndexChanged
       If Ready Then
           Performance.PortionValue = 0
           objdsAssessment.RejectChanges()
           For index = 0 To 10
               dataoption(index) = originalData(index)
           Next
           OptionWeight()
           SetListBox(ComboOptions.SelectedValue)
           GroupData.Text = "Orginal Datagrid Value"
           GroupData.ForeColor = Color.Blue
           LstBoxGrid.ForeColor = Color.Blue
       End If
   End Sub
```

' Change the slider value for selected alternative's datagrid value

```
   Private Sub Performance_Change(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Performance.Change
       objdsAssessment.RejectChanges()
       selOptRow = objdsAssessment.TblDatagrid.DefaultView.Count
       Dim i, j As Integer
       For i = 0 To selOptRow - 1
           If GridOption(i, 0) = ComboOptions.SelectedValue Then
               For j = 5 To 24
                   GridOption(i, j) = Round(GridOption(i, j) * (1 + Performance.PortionValue), 3)
               Next
           End If
       Next
       GroupData.Text = "Revised Datagrid Value"
       GroupData.ForeColor = Color.DarkGreen
       LstBoxGrid.ForeColor = Color.DarkGreen
       SetListBox(ComboOptions.SelectedValue)
       MenuCalDatagrid()
       OptionWeight()
   End Sub
```

' Change the threshold for selected alternative

```
   Private Sub GridThreshold_CurrentCellChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GridThreshold.CurrentCellChanged
       createChart(UtilityForm, GridThreshold)
       createTotal(UtilityTotal, GridThreshold)
   End Sub
```

' Delete all the change made in utility analysis

```
   Private Sub BtnReset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnReset.Click
       TxtBudget.Text = CStr(Budget)
       Dim selOption As Integer = GridThreshold.VisibleRowCount
       Dim i As Integer
       For i = 0 To selOption - 1
           GridThreshold(i, 1) = Round(Percent(i) * Budget * 10 / 7 * 0.5)
           GridThreshold(i, 2) = Round(Percent(i) * Budget)
           GridThreshold(i, 3) = Round(Percent(i) * Budget * 10 / 7 * 0.8)
           GridThreshold(i, 4) = Round(Percent(i) * Budget * 10 / 7)
       Next
       createChart(UtilityForm, GridThreshold)
       createTotal(UtilityTotal, GridThreshold)
   End Sub
```

' Change the current total budget

```
   Private Sub TxtBudget_KeyUp(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles TxtBudget.KeyUp
       If e.KeyCode = Keys.Enter Then
           createTotal(UtilityTotal, GridThreshold)
       End If
   End Sub
#End Region
End Class
```