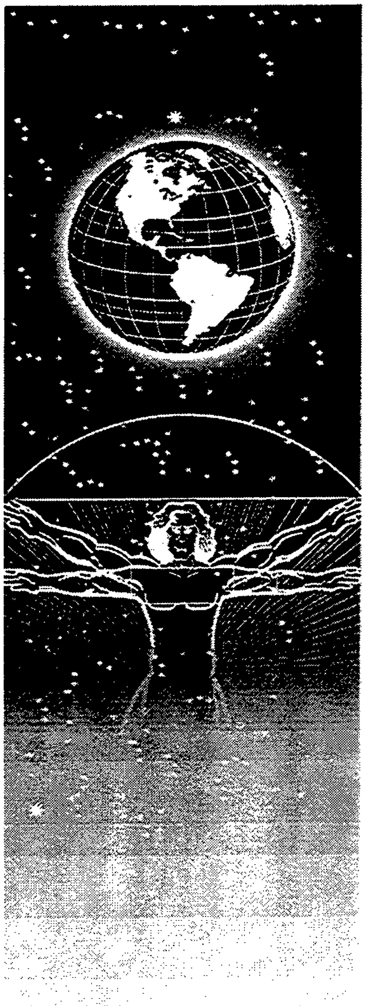
AFRL-HE-WP-TR-1998-0037



UNITED STATES AIR FORCE RESEARCH LABORATORY

Application of AHP to Requirements Analysis

Alexander Kott, Ph.D. and William Boag, Ph.D.

Carnegie Group, Inc.
Five PPG Place, Pittsburgh, PA 15222

Luis Vargas, Ph.D.

The Joseph M. Katz Graduate School of Business
University of Pittsburgh, PA 15260

April 1998

Final Report for the Period February 1993 to September 1996

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Crew Survivability and Logistics Division
Sustainment Logistics Branch
2610 Seventh Street
Wright-Patterson AFB OH 45433-7901

NOTICES

DISCLAIMER

This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-1998-0037

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

THOMAS J. MOORE, Chief
Crew Survivability and Logistics Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1998	Final - February 1993 - September 1996

4. TITLE AND SUBTITLE
Application of AHP Requirements Analysis

5. FUNDING NUMBERS
C-F41624-92-C-5001
PE - 62202F
PR-1710
TA-D1
WU-03

6. AUTHOR(S)
Alexander Kott, Ph.D. and William Boag, Ph.D., Luis Vargas, Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carnegie Group Inc.
Five PPG Place Pittsburgh
Pittsburgh, PA 15222

The Joseph M Katz. Graduate School of Business
University of Pittsburgh
Pittsburgh, PA 15260

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Crew Survivability and Logistics Division
Air Force Material Command
Sustainment Logistics Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-1998-0037

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
We describe a group of related approaches to quantitative evaluation and prioritization of system requirements and proposals. In formulating requirements for new systems, analysts face difficult tasks of selecting between alternative requirement decompositions, identifying questionable or low-priority requirements, eliminating requirements that add unnecessary costs, projecting cost impact of the requirements and then evaluating alternative system design proposals against the stated requirements. We seek approaches that require modest investments of time and can be applied even at the early stages of the requirements formulation process. We show that the Analytical Hierarchy Process can be applied to provide such evaluation mechanisms and illustrate our approach with a detailed example.

14. SUBJECT TERMS
RAPID; Requirements; requirements specifications; functionality; functional contracts; design constraints; design constraints data; communication protocols; project management information; system objective; cost impact

15. NUMBER OF PAGES
34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This report documents the results of a study of different approaches to quantitative evaluation and prioritization of system requirements and proposals conducted as part of a logistics research and development program titled Requirements Analysis Process in Design for Weapons Systems (RAPID-WS) (contract number F41624-92-C-5001), managed by the Air Force Research Laboratory, Logistics Sustainment Branch (AFRL/HESS), Wright-Patterson AFB, OH. The primary focus of this program was to evaluate approaches that require modest investments of time and can be applied even at the early stages of the requirements formulation process. The RAPID-WS system currently under development for the Air Force serves in this paper as a prototypical system for computer-aided requirements analysis.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

Requirements specifications for a system typically include functionality, functional constraints, design constraints, data and communication protocols, project management information, and system objectives, including the desired interactions with the eventual systems environment (McDermid, 1993). The primary objective of requirements analysis is to provide necessary and sufficient information such that the proposed system can be implemented successfully.

In formulating requirements for new systems, analysts face the difficult tasks of selecting between alternative requirement decompositions, identifying questionable or low-priority requirements, eliminating requirements that add unnecessary costs, projecting cost impact of the requirements, and then evaluating alternative system design proposals against the stated requirements. In this paper, we seek approaches that require modest investments of time and can be applied even at the early stages of the requirements formulation process. We show that the Analytical Hierarchy Process can be applied to provide such evaluation mechanisms and illustrate our approaches with a detailed example. The problems addressed in this paper include the following:

- Ranking of individual requirements in a requirements set interms of cost, risk, and benefit criteria.
- Evaluation of alternative requirements for a given systems need.
- Projection of systems costs based on requirements.
- Comparison of alternative systems proposals.

Although the approaches discussed in this paper can be applied to requirements analysis in diverse areas of systems applications, we developed them in the context of the requirements definition and management procedures of the United States Air Force, particularly within the framework of the project called Requirements Analysis Process in Design for Weapons Systems (RAPID-WS), sponsored by Human Systems Center (AFMC), Armstrong Laboratory, Logistics Research Division, Wright-Patterson AFB OH. For this reason we will occasionally refer to the terminology and procedures accepted in the US Air Force. The RAPID-WS system currently under development for the Air Force serves in this paper as a prototypical system for computer-aided requirements analysis.

1.1 The Role and Process of Requirements Formulation

The development and evaluation of requirements for large systems are complex and labor-intensive processes focused on the needs of end-users. It typically starts by identifying a few high-level goals that the system is expected to accomplish. For example, in US Air Force practice, the process of developing operational requirements starts with identification of high-level goals called mission needs. The specification of a mission need is based on the identification of existing system deficiencies and constraints, the existence of external threats not addressed by current systems, and the potential for the creation of new solutions to satisfy a need.

Organizations use a variety of procedures, some of them highly structured, to identify and validate such high-level goals or mission needs. The high-level goals are then refined or decomposed into progressively more detailed operational and other requirements for the proposed system. The processes of requirements identification, decomposition and validation must include extensive input from the end-user as well as input from the system experts.

Requirements documents are commonly generated and documented as a hierarchy of requirements, because the specification of requirements is a difficult task for human problem solvers, involving the manipulation of large and diverse units of knowledge (Vessey and Conger, 1994). Complex tasks are often addressed by decomposing the problem into a hierarchy of subproblems for which solutions can be found or generated (Simon, 1962). The key to this approach is to restructure the problem into subproblems such that the subproblems can be integrated into a hierarchy which forms the solution to the complete problem. In the practice of requirements definition, this decompositional approach takes the form of a hierarchical formulation of requirements. For example, Table 4 contains a partial hierarchy for a transportation planning system which will be discussed in Section 6.

The quality and effectiveness of the requirements formulation processes are of extreme importance. A system built to meet a set of requirements will be able to meet the true needs of the end-users only as well as those needs are captured in the documented requirements. The life-cycle costs of the system will be critically impacted by the requirements as well; unnecessary or excessive requirements can have a dramatic impact on the system costs.

Once developed, system requirements are used for a variety of purposes. They serve as a basis for projecting and evaluating the cost and operational effectiveness of proposed systems. They are also used as a tool for solicitation and evaluation of alternative proposals to construct and/or operate the system. Once a particular proposal is selected, requirements are used as a starting point for detailed design and for controlling the success of the system development project.

1.2 Computer-Aided Requirements Analyses

Computer-aided analyses have been motivated by the difficulties encountered in the definition of requirements using complex manual management system procedures. Challenges of the requirements definition and analyses include among others:

- Individual requirements can be incompletely defined.
- A requirements set can be incomplete.
- Requirements may be insufficiently decomposed.

- Requirements may be ambiguous.
- Cost and risk implications of requirements can be difficult to predict and understand.
- Ranking of alternative requirements sets from diverse sources and multiple areas of expertise is difficult.
- The relative merit of requirements with respect to cost, operational effectiveness, or other criteria can be difficult to determine.
- Using requirements to evaluate proposed solutions is difficult.

The techniques proposed in this paper were developed while working on the design of a software tool for requirements analysis which addresses a number of these challenges (Kott and Peasant, 1995). In this paper we focus on the quantitative techniques that address challenges involving ranking of individual requirements, sets of requirements and proposed systems.

1.3 The RAPID-WS System

The techniques proposed in this paper originated in our work on the development of the Requirements Analysis Process in Design for Weapons Systems (RAPID-WS) - a computer-aided requirements engineering tool for the creation and management of operational requirements for the United States Air Force (Kott and Peasant, 1995; Popken and Perkins, 1996). RAPID-WS provides facilities for semi-formal object-oriented representation of individual requirements, hierarchical and non-hierarchical relations between requirements, facilities for concurrent collaborative development and management of requirements, mechanisms for enforcing certain aspects of completeness and consistency of requirements hierarchy, capabilities for version management, and facilities for semi-automated generation of variety of documents from an underlying database of requirements.

The RAPID-WS project is sponsored by the by the Human Systems Center (AFMC), Armstrong Laboratory, Logistics Research Division, Wright-Patterson AFB, OH. It is expected that some of the approaches presented in this paper, particularly the approach to comparison of alternative requirements, will be implemented within the RAPID-WS, and a detailed software design has been developed for such an implementation.

1.4 Multicriteria Decision Making Techniques

A critical task for a decision maker may include the evaluation of alternative requirements to select a system for implementation among a group of alternative systems solutions. Another task may be to discriminate among a set of individual requirements relative to cost, risk, or other criteria to determine the optimal requirements to satisfy a mission need. Multicriteria Decision Making techniques are used to help a decision maker make a choice among a set of pre-specified alternatives. MCDM techniques allow the analyses of multiple evaluation criteria and the incorporation of the decision maker's preferences on these criteria into the analyses. These techniques are necessary since it is difficult for a decision maker to perform manual evaluations in situations where the set of criteria and set of alternatives are large.

A variety of different MCDM alternative evaluation techniques have been developed and published in the literature (Karni, et al. 1990; (Narasimhan and Vickery, 1987; Korhonen, et al, 1992; Reitveld and Ouwersloot 1990; Schuijt 1994). The well known and widely used Analytical Hierarchy Process (AHP) (Saaty, 1980; Saaty, 1986) is particularly suited to the evaluation of alternative, hierarchical requirements sets since the process enables the user to develop a hierarchy of criteria to be used as a basis for the evaluation of a set of alternatives.

The AHP is used to derive priorities in multicriteria decision making. It is based on three principles: Decomposition, Measurement of Preferences, and Synthesis. Decomposition breaks a problem down into manageable elements that are treated individually. It begins with implicit descriptors of the problem (the goal) and proceeds logically to criteria (or states of nature) in terms of which outcomes are evaluated. The result of this phase is a hierarchical structure with levels for grouping issues together that are of homogeneous importance or influence with respect to the elements in the adjacent level above. A ratio scale of measurement is derived from pair-wise comparisons of the elements in a level of the hierarchy with respect to the influence of an element in the level above. Pairwise comparisons are made, with judgments provided as verbal statements about the strength of dominance (importance or likelihood) of one element over another represented numerically on an absolute scale. These judgments are made in the framework of a matrix used to derive a local priority vector as an estimate of relative magnitudes of the elements being compared. When priority vectors are derived for all comparisons in the hierarchy, one proceeds to synthesize the local priorities to derive a global measure of priority used to make the final decision. These global priorities are obtained by successively weighting and adding from the top level to the bottom level of the hierarchy. The outcome of the synthesis is a multilineal (and hence nonlinear) form whose complexity depends on the number of elements in each level and on the number of levels in the hierarchy.

A particular advantage of the AHP approach is that the criteria used for the evaluation of the alternatives may be arranged in a hierarchy. This hierarchical model allows the user to make simple pairwise comparisons of relative importance among a fairly small set of criteria at each level in the hierarchy. This minimizes problems encountered with some of the other evaluation techniques, such as Multiattribute Utility Theory (MAUT) (Keeney and Raiffa, 1976), where a large number of alternative evaluation criteria, assigned to a single, flat level, must be compared with each other; or Outranking Methods such as Electre II (Roy, 1968; Roy and Bertier, 1973) which in addition to the aforementioned problems, only produce a dominance graph, i.e. the non-dominated alternatives are identified and represented using a directed graph. In addition, MAUT cannot deal with more than one level of complexity because the scales used are interval scales. Outranking methods generate ordinal scales and share the same problem as MAUT with the additional concern that in most of the cases their criteria do not have weights assigned to them. The outranking relation used to construct the directed dominance graph does not require more than an ordinal scale. The problem with this is that in many situations no an alternative may be clearly dominant. For a comparison of the three methodologies see Vargas (1994).

AHP's ability to deal with hierarchies is a good match for the hierarchical nature of requirements decompositions. In the context of requirements evaluation we propose to apply the AHP approach for the following purposes:

- Ranking of individual requirements in a requirements set in terms of cost, risk, and benefit criteria.
- Evaluation of alternative requirements for a given systems need.
- Projection of systems costs based on system requirements.
- Comparison of alternative systems proposals.

1.5 Organization of the Paper

This remainder of the paper is organized as follows. In Section 2 we describe an approach to the prioritization of requirements in terms of cost, benefits, and risk. In Section 3 we discuss a method which allows hierarchical requirements to be used as a basis for forecasting system costs. In Section 4, we apply the Analytical Hierarchy Process (AHP) method to the evaluation of alternative proposals. Section 5 illustrates the application of AHP to the task of comparing several alternative sets of requirements meeting the same need. In Section 6 we develop a detailed example of the application of the proposed techniques to evaluation of requirements, projection of costs and development of alternative design concepts for a transportation planning system.

2.0 Comparison of Requirements: Cost, Benefit, Risk

A requirements analyst is often faced with a need to rank a set of requirements with respect to criteria such as system cost, operational benefits or developmental risks. For example, a design concept study may indicate that the system satisfying a given set of requirements will be too expensive. The analyst is asked to identify the requirements that can be eliminated in order to reduce the cost of the system. In current practice, the analyst does not have a formal tool identifying a subset of requirements that could be eliminated in order to reduce the cost while minimizing the negative impact on operational effectiveness. In another example, the program management desires to identify high-risk requirements in order to apply risk-mitigating measures. The analyst is requested to identify such high-risk requirements.

Specific criteria relevant to such rankings may vary. In one case the analyst may be concerned about the initial acquisition cost, in another case, about the program risk. However, in general such criteria tend to fall into one of three categories: costs, benefits, risks.

To address the need for requirements ranking we propose to exploit the hierarchical nature of requirement sets by applying the AHP methodology. Consider a hypothetical hierarchy of requirements in Figure 1. We begin by pair-wise comparison between the upper-level requirements R1 and R2 with respect to the question "which requirement contributes greater cost to the overall cost of the system?" Suppose this comparison results in R1 being given a measure of .2 and R2 being given a measure .8. Similar comparison between requirements R21, R22 and R23 is done with respect to the question "which requirement contributes greater cost to the satisfaction of requirement R2?" This comparison results in measures .1, .4 and .5 respectively. Multiplied by the measure associated with R1 itself, it leads to measures .08, .32 and .40 respectively. The subrequirements of R1 are treated similarly. The intuition here is that the measure associated with each requirement corresponds to the fraction of the overall system cost attributable to this requirements. For example, requirement R11 causes 6% of the overall cost, R22 causes 32% of the cost, etc.

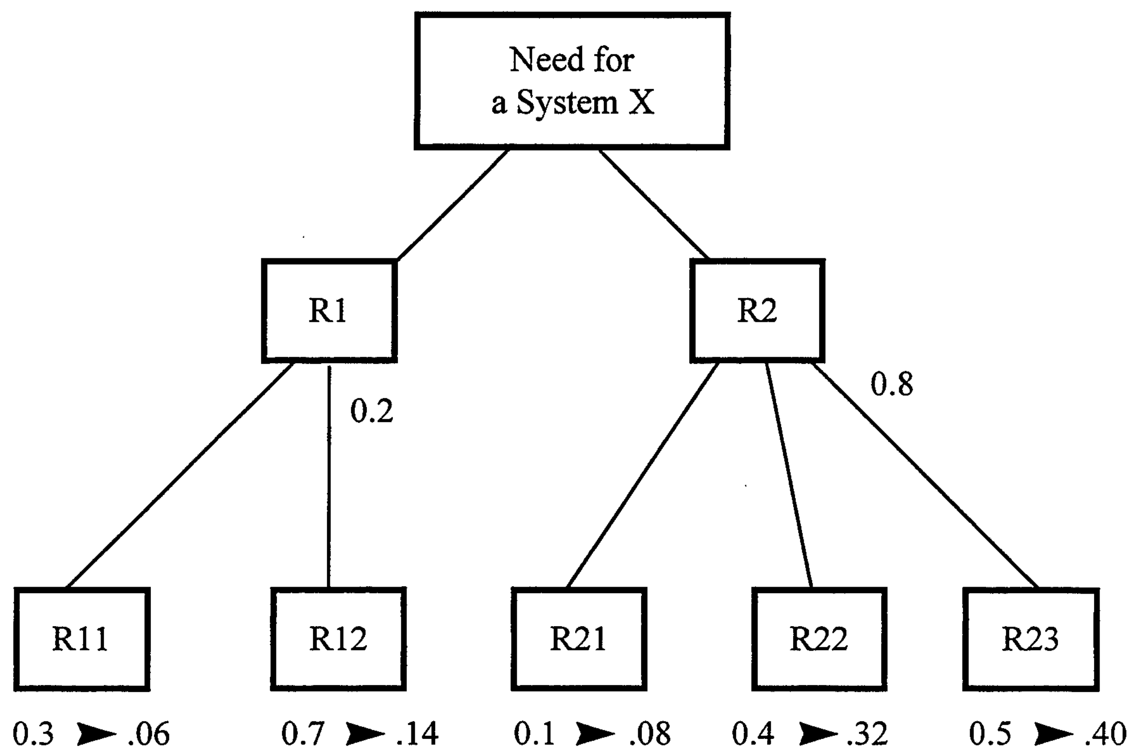


Figure 1: Ranking in a hierarchy of requirements.

This way of propagating weights down a hierarchy is based on the assumption that the elements under R1 and R2 are only dominated by R1 and R2. No other dependencies are assumed. If that were the case, then one would use an extension of the principle of hierarchical composition known as the supermatrix approach (Saaty, 1980). A recent book by (1996) gives more details of the potential applications of this extension, now known as the Analytic Network Process. If it is assumed that the hierarchies are independent, it can be shown that the principle of network composition becomes the principle of hierarchical composition (Saaty, 1980). A simple example of the validity of this approach is an input-output Leontieff matrix of an economy (Saaty and Vargas, 1979). In this case, a basic assumption is that a sector output is proportional to the contributions (inputs) from the other sectors of the economy. A similar assumption is made in the AHP. Without that assumption the problem should be treated as a non-linear hierarchy or possibly as a network in general.

The resulting numbers (Figure 1) can be very instructive. The analyst may want to review outliers, e.g., R11 or R23, more closely and adjust some of the judgments. The analyst may also want to review R22 and R23 in detail and see if these requirements can be modified in a manner that would make them less costly. It is also possible that the relatively high cost measure reflects misunderstanding of the requirements, i.e., "reading too much" into them. In such a case, merely a reformulation of the requirement statement or a more elaborate definition may help to eliminate the misunderstanding and the associated overestimate of the relative cost impact. Also, these two requirements may need to be decomposed further and stated with greater precision given their heavy cost impact.

This approach resembles the common practice of analyzing the costs of a system by decomposing the system into a hierarchy of subsystems, modules, etc., and then estimating the cost of individual modules. For example, in software systems such a decomposition is a part of architectural design. Unlike the approach we propose here, this conventional practice requires making a significant number of decisions regarding the design concept of the system and also depends in a complex way on the formulation and decomposition of requirements (e.g., Booch, 1986; Hester, 1981; Yourdon, 1985)).

The important advantage of our approach is that it can be applied during the requirements definition phase, prior to making any decisions about the design of the system, its architecture, decomposition into subsystems, allocation of functions for subsystems, or even about the basic design concept. In many cases, it can be completed by analysts whose primary expertise is not in the system design but in operational aspects of the future system and who are likely to be closer to the needs and requirements of the end-users. This does not eliminate the need for the analyst to have experience and insights regarding the relations between the nature of requirements and their likely cost impacts.

There is serious concern evidenced here. In a system-subsystem decomposition, subsystems or modules generally do not overlap, with the relatively minor exceptions of interface components. In the requirements decomposition, the requirements are likely to overlap in the sense that meeting one requirement may allow another requirement to be met at a reduced cost. For example, if the cost of meeting requirement R1 alone is C(R1), the cost of meeting requirement R2 alone is C(R2) and the cost of meeting both requirements together in a single system is C(R1, R2), it is possible (but not necessary) that:

$$C(R1) + C(R2) > C(R1, R2)$$

Suppose requirement R1 is to provide a user of a system with the means to enter and save certain data. R2 is a requirement to provide the user with the means to retrieve the same data. Both may imply a need to provide a database, and very likely the same database will support both requirements. If our analyst considers the impact of these two requirements separately, she will "double-count" the costs implied by the provisioning of the database.

To deal with this "double-count" Saaty developed the Analytic Network Process (ANP) (Saaty, 1980). An example of its application can be seen in Saaty and Takizawa (1986). The "double-count" problem mentioned here is a very specific case of a hierarchy with dependence among the elements of a level. To derive the correct priorities for the elements one first derives priorities for the elements if they were independent (independence priorities). Then, after identifying the relationships among the elements, one evaluates the impact of each element on the other elements, resulting in a square matrix with as many columns as elements being independently compared (matrix of influences). Finally, this matrix of influences is multiplied by the vector of independence priorities: (matrix of influences) x (independence priorities).

The result is the vector of dependence priorities. A simple example is given in Saaty and Vargas (1982). An easier way of dealing with this double-counting problem is to redefine the criteria in such a way that no double-counting takes place. This requires practice but it can be easily done.

From the practical perspective, this may not be as serious a limitation as it may appear. In applying this approach to practical problems, we found that analysts tend to recognize the possibility of double-counting, and deal with it by explicitly stating their assumptions and assigning shared costs to only one of the requirements. In the example we described above, an analyst would assume that the costs of providing a database are associated with R1, and R2 will be satisfied by taking advantage of the existing database. We will return to this concern in Section 6 where we describe a practical example.

Operational benefits associated with individual requirements can be analyzed in a similar manner. Let us refer to the same Figure 1, but this time we interpret the numbers shown in the figure as measures of operational benefits.

We begin by pair-wise comparison between the upper-level requirements R1 and R2 with respect to the question "which requirement provides greater benefit?" This comparison results in R1 being given a measure of 0.2 and R2 being given a measure of 0.8. Similar comparison between requirements R21, R22 and R23 is done with respect to the question "which requirement contributes more to the benefits provided by satisfaction of requirement R2?" The numbers associated with leaf nodes can be interpreted as measures of benefit contributed by each requirement. For example, requirement R11 provides 6% of the overall benefits, R22 provides 32%, etc.

The same technique can also be applied to analyzing the risk associated with requirements. For example, development of software systems involves risks of financial failures (time and budget overruns) and technical failures (failure to meet functional and other requirements). Effective risk management requires the ability to rank and prioritize various risk factors involved in the project. Boehm (1989) offers a number of checklists to help identify risk factors. Use of such checklists requires a significant number of decisions or assumptions about the system's design and about the approach to project organization and execution. Ranking of risks in approaches such as Boehm's relies on estimating probability of failure and the loss associated with the failure. Both numbers are difficult to estimate.

In contrast, our approach relies exclusively on the hierarchy of requirements and the information available at the time of requirements definition. Let us consider a particular risk that tends to be of primary concern in evaluating requirements for a new system: the risk that a given requirement will introduce difficulties (technical or programmatic) that will prevent the project from being completed on time and within budget. Given this definition of risk the analyst can use the AHP method in the same way we discussed cost and benefits, by answering questions such as "how much more risk is introduced by R1 as opposed to R2?" Let us once again refer to the same Figure 1 but this time we interpret the numbers shown in the figure as measures of risk.For Example, requirement R11 contributes 6% of the overall project risk, R22 contributes 32%, etc.

Even though the definition is admittedly ambiguous, our experiments suggest that a requirements analyst who has had experience with projects involving somewhat similar requirements tends to have a rather definitive opinion on relative risks of requirements. Furthermore, judgments of multiple analysts are generally consistent (Section 6).

Having discussed the way to determine relative cost, benefit and risk impact of requirements, we can now suggest use of a composite measure proposed by Saaty (1987). Saaty combines cost, benefit and risk measures as b/(c*r), where b is the measure of benefit, c is the measure of cost and r is the measure of risk.

Now broadly used, this measure was developed by Saaty to cope with situations in which probabilities could not be used because of the ambiguity involved. One can interpret this composite measure as an amount of benefit weighed by risk per unit of cost. A requirement with higher composite measure is more desirable than the one with lower composite measure. The value of the composite measure can be obtained by first computing cost, benefit and risk measures using the AHP and then simply computing a b/(c*r) measure for each requirement. Figure 2 is an example where all measures are computed for every requirement in the hierarchy.

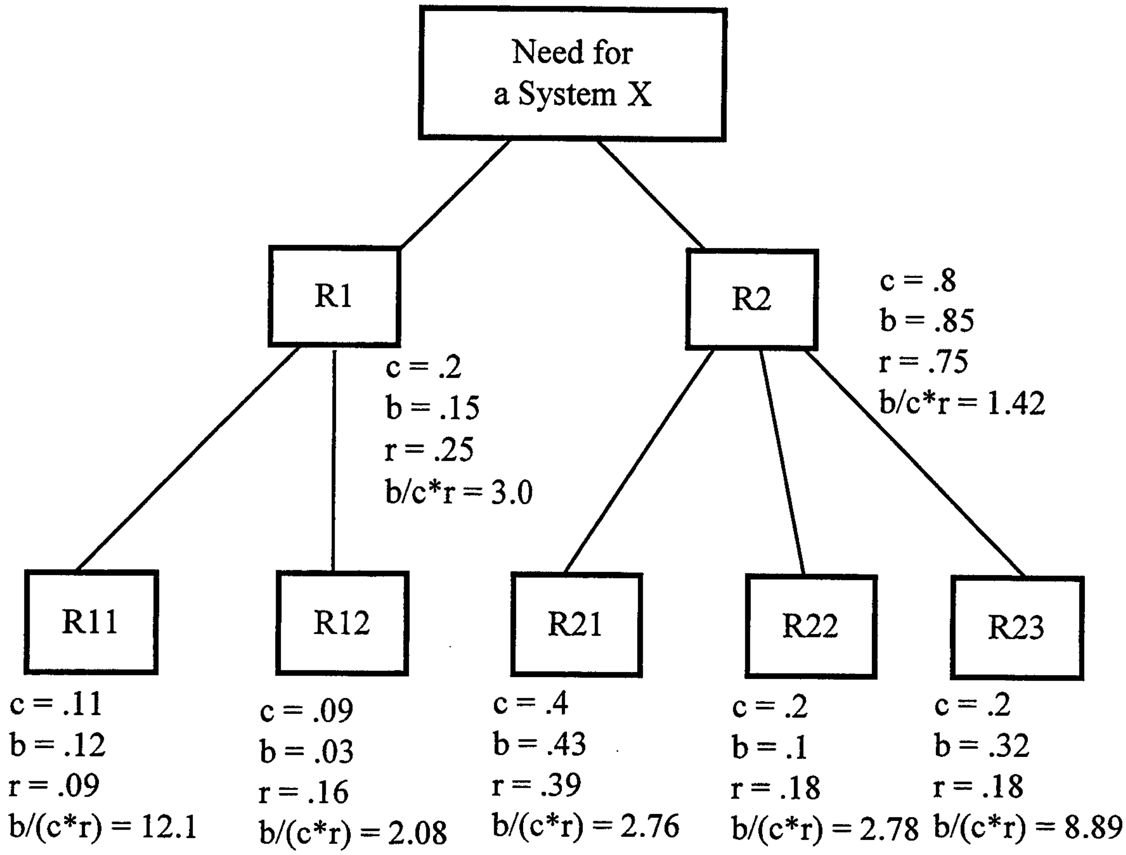


Figure 2: Cost, benefit, risk and the composite b/c*r measures.

Just as we suggested in the case of cost, the analyst can review the values of composite measures from several useful perspectives:

- Are outliers justifiable? Is a very high (or very low) value of b/(c*r) a true reflection of the nature of the requirement, or is it an indication of an error in estimating values of cost, benefit or risk? Perhaps the analyst should return to an earlier step of the process and modify the earlier judgments.

- If a requirement has a very low b/(c*r) measure, can it be eliminated? Modified? Replaced by an alternative requirement(s) that would meet the need of the end user at lower cost or with lower risk?

- If a requirement has a very low b/(c*r) measure and the cause is high risk, is it possible to apply programmatic measures of risk mitigation?

- Perhaps relatively low benefit is an indication that the requirement is of a supporting nature: it does not bring directly any operational benefits but is introduced because the analyst considers it important in support of another requirement. In this case, it may not be a true requirement (i.e., it is an implementation issue) and it may be appropriate to eliminate it.

- If a requirement has a very high b/(c*r) measure, is it because it has not been sufficiently decomposed?
- If a requirement has a very high b/(c*r) measure, would it be advantageous to emphasize this in program directives to the vendors? For example, require them to give priority to these high-value requirements early in the project?

3.0 Forecasting Cost of the Project in the Requirements Phase

Let us return to the discussion of cost analysis we started in the preceding section. Until now we focused on determining relative cost impacts of requirements. In this section we discuss how this analysis of relative costs can be extended to estimating the overall cost of the system.

Techniques of cost estimating for system projects often include as the first step decomposition of the proposed system into a hierarchy of subsystems, modules, etc. The cost of leaf-level entities (e.g., components) is estimated by estimating a "size" measure of each entity and applying an experience-based metric that relates size and cost. For example, in software system engineering, cost estimating practices generally fall into one of two classes of approaches: one relies on the ability of the cost analyst to estimate the size of code in lines (the COCOMO model, Boehm (1984)), the other relies on size estimates in terms of function points (Albrecht and Gaffney, 1983; Jones, 1986). Both classes of approaches require making a large number of system design decisions and a significant investment of effort into detailing the design to the point where the size of individual modules or number of function points can be estimated with adequate accuracy.

We are interested in formulating an approach that can be applied at the requirements analysis stage and prior to making design and implementation decisions. In the software estimation field, it is often claimed that techniques based on function-point approaches are applicable at the requirements phase, as they require as input only user-defined requirements and do not require design information. However, use of function-point techniques does require rather detailed requirements that frequently involve implied design decisions. For example, the cost analyst needs to know or to assume the number of reports required, the number of fields within each report, etc. In both Government and commercial practices, such a level of detail is not available when operational users are defining requirements.

It would be desirable to develop an approach that can be applied very early in the requirements development cycle and which is relatively insensitive to the level of detail at which the requirements are available. The example in Section 6 illustrates an application of our approach to a hierarchy of requirements that does not have the degree of detail that the function-point approach requires.

The approach we propose here consists of the following steps:

1. Formulate a hierarchy of requirements.

2. Estimate the relative cost impact c(R) of every requirement R by applying AHP as described in Section 2.

3. Select a subset {Ri} of leaf-level requirements.

4. Estimate costs {C(Ri)} required to fulfill requirements within the selected subset.

5. Compute the total cost of the system as **SUM** (C(Ri))/**SUM**(c(Ri)).

Let us illustrate this process using the hierarchy of requirements in Figure 1, and assuming that the measures shown next to each requirement represent relative costs. These measures are determined using AHP as discussed in Section 2. For example, requirement R12 contributes 14% to the overall cost of the system and requirement R21 contributes 8%. Suppose we have reasons to believe (perhaps based on our prior experience with a somewhat similar system where two similar requirements were met) that requirement R12 would contribute $20,000 to the overall cost of the system and requirement R21 would contribute $10,000. We can now estimate the overall cost of the system:

($20,000 + $10,000) / (.14 + .08) = $136,364

In practice, there are significant challenges associated with estimating costs {C(Ri)} required to fulfill requirements within the selected subset. There are several ways for an analyst to approach the task of estimating the actual cost contribution of a given requirement, including:

1. Request system design experts to perform a cost estimate for the requirement, using conventional cost estimating approaches, such as performing a partial design of the modules required to satisfy the requirement and estimating costs of the individual modules.

2. Find cost data for similar requirements satisfied in systems already constructed. Care has to be taken to confirm that the differences between those systems and the system under consideration do not impact the cost of satisfying this requirement.

3. Use expert judgments to estimate the costs. This often also involves experts, implicitly or explicitly, finding and applying cost data from prior comparable experiences.

The first approach is likely to be the most expensive. The other two are less expensive and probably should be used in combination.

Other challenges are found in selecting an appropriate subset {Ri} of leaf-level requirements. How large should this subset be? On one hand, the larger the subset selected, the more accurate the estimates would be (errors in estimating individual requirements are more likely to cancel out). However, a larger subset will also require greater efforts to produce and estimate the requirements. Should this subset be selected randomly? It may be desirable to select only those requirements for which we can find data from prior experiences. However, such a selection can introduce serious biases and lead to less accurate estimates.

Is it desirable that the subset include requirements of different cost levels, e.g., some costly requirements and some inexpensive? In practice, it may be easier to deal with inexpensive ones. Could it impact the accuracy of the estimates?

At this time we do not have answers to the above questions. They are topics for further research. It should also be noted that the approach proposed here does not provide a mechanism for determining the accuracy of the cost estimate, e.g., a confidence interval. It appears that one way to approach these issues would be through a probabilistic analysis of the cost estimates. Such an approach would treat a cost estimate not as a single data point but rather as a probability distribution of the expected costs for each requirement within the hierarchy of requirements. A distribution for the overall cost could be constructed from the distributions of lower-level requirements. Using such an approach, the analyst may be able to fine-tune the selection of a requirements subset based on the resulting confidence interval. We propose this as a possible direction for future research.

In Section 6 we describe our approach to these issues in practice.

4.0 Comparison of Alternative Solutions

Hierarchical analysis of requirements in terms of cost, benefit and risk discussed in section 2 can have yet another important application: it can be used to evaluate alternative solutions proposed to meet the requirements.

Suppose we solicit proposals to build a system that would meet the hypothetical requirements hierarchy depicted in Figure 2 together with the relative cost, benefit and risk measures. In response to our solicitation we receive two proposals from two vendors: System-1 and System-2. Figure 3 depicts the AHP-based approach to evaluating these two proposals using the relative measures we already determined for our requirements:

1. For each requirement, we perform pair-wise comparison of System-1 and System-2 in terms of cost. For example, for requirement R11 we fill out a pair-wise comparison matrix with respect to the question "how much more (or less) costly will it be to implement requirement R11 within System-1 than within System-2?" This process results in the assignment of a relative cost measure with respect to R11 to both System-1 and System-2. In this way, for each requirement R and each proposal P we

form a cost measure c(R,P) that reflects our judgment regarding the cost of satisfying requirement R within proposal P.

2. For each alternative proposal, compute the overall cost measure as a sum of cost measures for all requirements: C(P) = **SUM**(c(R,P)).

3. In a similar manner, determine benefit and risk measures for each proposal.

4. Compute b/(c*r) measures for each proposal. These can be used as part of the input in the final selection of the winning proposal.

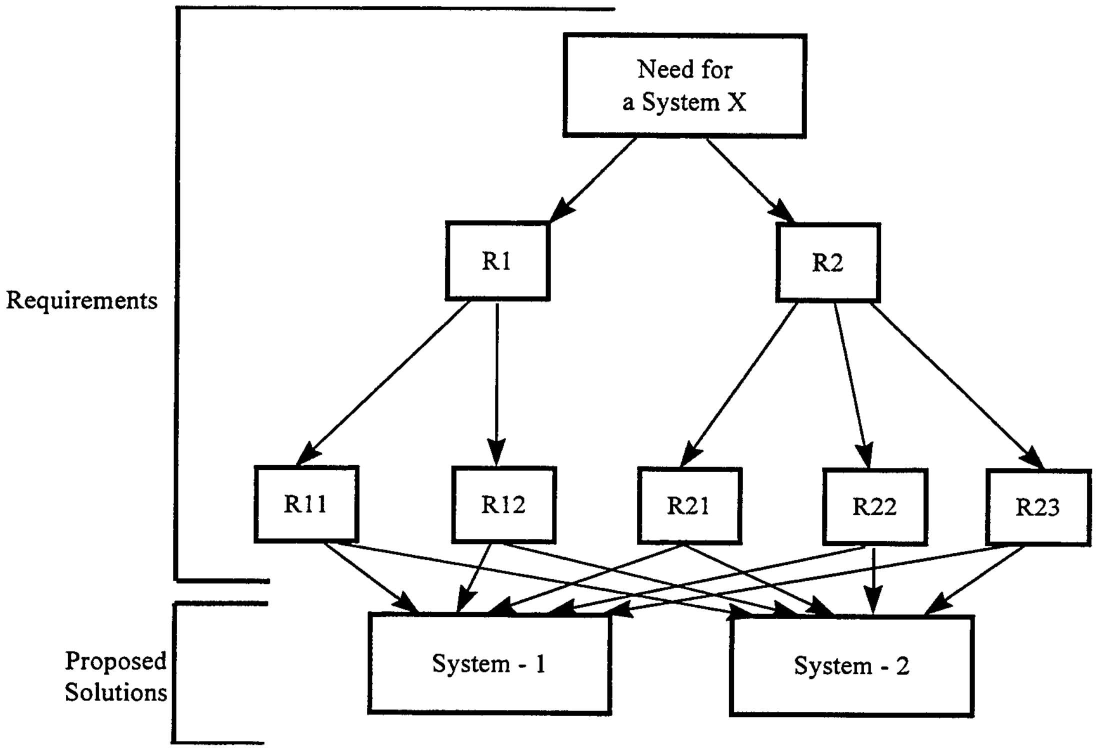


Figure 3: Proposed solutions are compared in terms of their ability to satisfy each of the leaf-level requirements. The sum of contributions from all requirements is the overall ranking of the solution.

In both Government and commercial practices, requirements formulation precedes solicitation of proposals, which is followed by formal evaluation and comparison of proposals. The analysis of requirements discussed in Section 2 can be accomplished prior to solicitation and comparison of requirements. The results of this analysis can be seen as a document that defines rigorously the criteria for comparative evaluation of proposals. It also becomes an approach and a tool for performing such comparison in a objective manner. We perceive these uses of the proposed approach as particularly important.

The relative measures of the requirements should be communicated to the prospective vendors as part of the proposal information package or RFP (Request for Proposal). Armed with this information, a vendor will be able to optimize various system concepts and select for the proposal those concepts that best satisfy of the requirements as defined by the relative measures.

To summarize, there are three important uses for the analysis of relative cost, benefit and risk measures of requirements in analyzing alternative proposals for solutions:

1. To formulate and document the proposal evaluation criteria and method prior to soliciting proposals.

2. To communicate to the prospective vendors the objective function with which vendors can optimize the proposed solution.

3. To enable the soliciting organization to rank proposals in a rigorous, formal and relatively objective manner.

5.0 Comparison of Alternative Requirement Sets

A common need arising in formulating requirements is to select between alternative sets of requirements. The problem is particularly acute in a concurrent requirements engineering environment where multiple operational and system experts may collaborate in developing a requirements set. For example, one analyst may propose a set of requirements that in her view address a particular need. Another analyst may feel that the same need implies a different set of requirements (Figure 4). A similar choice between competing sets of requirements may have to be made even when only one analyst is developing requirements for a system. In general, it is a problem of selecting between alternative decompositions in a hierarchy of requirements. We are interested in a prioritization technique that enables the analyst to perform such a selection. We call the process of prioritization "requirements alternatives evaluation."

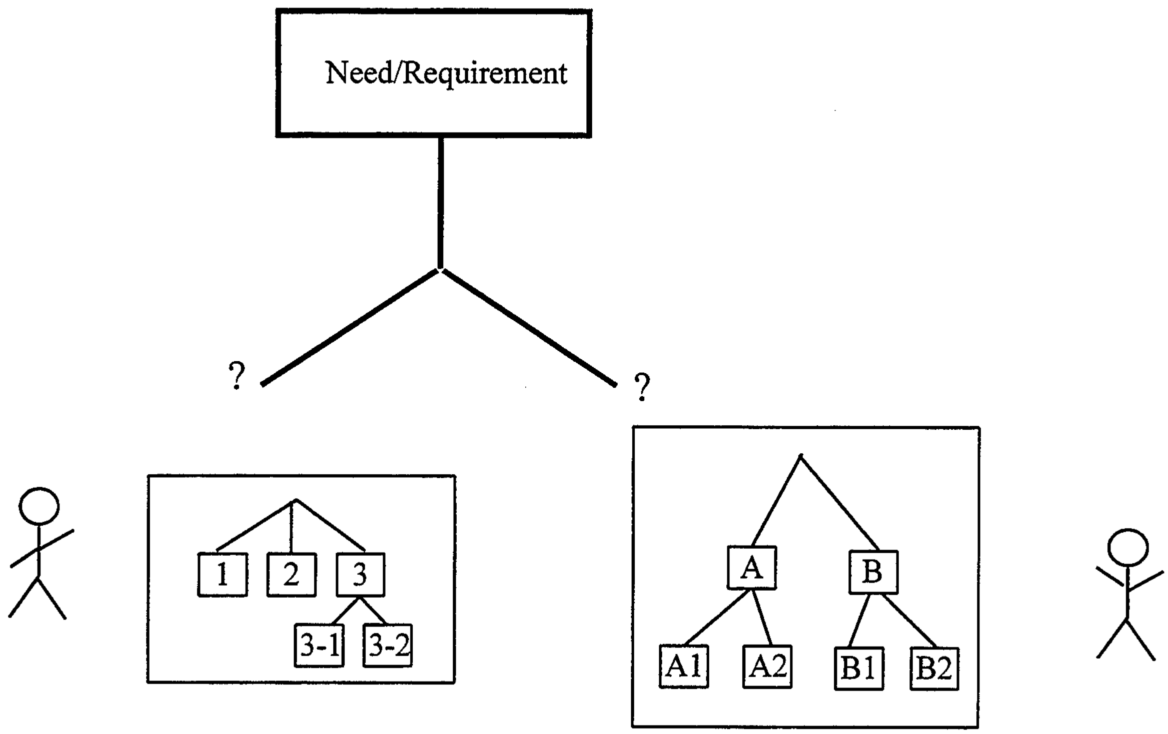


Figure 4: There may be different opinions on how a given need/requirement

One of the interesting issues in devising such an alternative evaluation technique is how to identify the criteria for selection. The alternative evaluation process is dependent upon the specification of a set of criteria which are to be used for the evaluation of the alternatives. Using manual methods, these criteria must be identified by the analyst. In a computer-aided system such as RAPID-WS, the criteria may be suggested by the system. We refer to the process by which the software system identifies candidate criteria as "criteria extraction."

We developed and designed for implementation in the RAPID-WS system an algorithmic criteria extraction procedure that determines a set of suggested evaluation criteria from the alternative requirement sets. The key idea is that the best source of criteria for comparing requirements are the requirements themselves.

The paper by Kott and Peasant (1995) discusses the representational approach adopted in the RAPID-WS system where many requirements can be represented as a couple <OBJECT PROPERTY RESTRICTION>. An object can be a system, a module, an operator, an environmental element, or a function. For example, in the requirement "The capacity of the main cargo bay will be at least X cubic meters," capacity is a property of the object "main cargo bay" and "at least X cubic meters" is the restriction. We find that when we compare two sets of requirements both of which are intended to meet the same need (or higher level requirement) such properties within requirements are strong candidates for comparison criteria. For example, the property "cargo bay's capacity" is a likely candidate for comparing two sets of requirements that include demands on the capacity of the cargo bay. This offers an opportunity: an automated system can suggest to the analyst a set of candidate criteria simply by extracting properties from the requirements being compared. This criteria extraction algorithm can be summarized as follows:

1. Extract properties of requirements from both sets of requirements (in general there can be more than two sets under comparison); in the case when the sets are hierarchical, perform this extraction to a predefined depth within the tree; form two sets of properties.

2. Form an intersection of the two property sets; the intuition here is that only those properties that are found in both sets are likely to be useful for the purposes of comparison;

3. Present the selected properties to the analyst, who can at that point select a subset of the suggested criteria or add new criteria.

The process of evaluating alternative sets of requirements consists of the following steps:

1. Determine the decision issue.

2. Select the evaluation criteria.

3. Construct the hierarchy of decision elements.

4. Assign alternative and criterion-relative weighting factors.

5. Calculate alternative evaluation rankings.

6. Evaluate the ranking results.

Let us illustrate the process of evaluating two alternative sets of requirements using the following example of a typical three level decision hierarchy model. This example consists of a mission need, a flat level of three criteria, and two alternatives representing two sets of requirements (Figure 5)

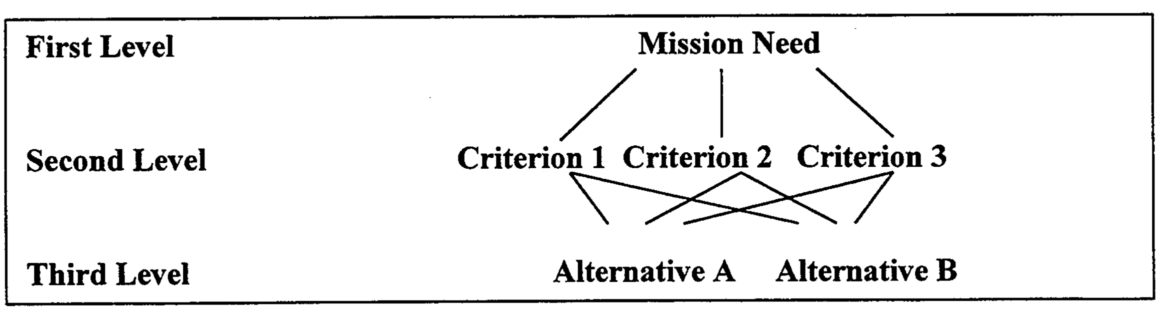


Figure 5: A typical three-level decision hierarchy model.

In this example, the analyst selects the requirements set for each alternative and starts the evaluation. At this point, a criteria extractor examines the requirements sets and extracts evaluation criteria from these requirements. These criteria and the values of the criteria with respect to each alternative are presented to the analyst in tabular form (Table 1).

Evaluation Criteria	Alternative A	Alternative B
Criterion 1	Rel. Value A to 1	Rel. Value B to 1
Criterion 2	Rel. Value A to 2	Rel. Value B to 2
Criterion 3	Rel. Value A to 3	Rel. Value B to 3

Table 1: Relative values of criteria with respect to alternatives.

This table lists all of the extracted criteria. An analyst may discard any of the criteria which are judged to be irrelevant and may enter additional criteria which are to be used in the evaluation. Assume that the analyst wishes to use all of the evaluation criteria suggested by the system and does not wish to add any additional criteria. The analyst accepts the suggested evaluation criteria for the evaluation process.

Next, the analyst assigns the final selection of relative weights comparing the elements in a pairwise fashion. These weights reflect the analyst's view of the relative contribution or impact of each element on its governing element in the next higher level. Tables may be provided which enable the analyst to assign relative weights as a measure of the importance of each criterion with respect to each alternative and the relative importance of each alternative with respect to each of the other alternatives in accomplishing the Mission Need.

The second level criteria are compared in pairs with respect to the mission need in the first level. This establishes the relative importance of each criterion with respect to the mission need. Table 2 shows the pairwise comparison matrix of the criteria with respect to the mission need with assumed values for the relative weights.

Mission Need	Criterion 1	Criterion 2	Criterion 3
Criterion 1	1	Rel. Value 2 to 1	Rel. Value 3 to 1
Criterion 2	Rel. Value 1 to 2	1	Rel. Value 3 to 2
Criterion 3	Rel. Value 1 to 3	Rel. Value 2 to 3	1

Table 2: Pairwise comparison of criteria to mission need.

Next, three comparison matrices are constructed for comparing the two alternatives with respect to each criterion as shown in Table 3. The comparison matrices for Criterion 2 and Criterion 3 are similar.

Criterion 1	Alternative A	Alternative B
Alternative A	1	Rel. Value B to A
Alternative B	Rel. Value A to B	1

Table 3: Pairwise comparison of alternatives with respect to criterion 1.

In the next step, the analyst performs calculations for the evaluation of the alternatives. Alternative sets of requirements for a specified mission need are computationally compared, based on analyst-designated sets of evaluation criteria, relative criteria weights, and relative alternative weights. Ranked numerical priorities are assigned for the set of available alternatives. The alternative which has the highest numerical ranking factor best meets the needs for the decision issue, relative to the chosen evaluation criteria.

The analyst can review the overall ranking of the alternatives, execute any desired sensitivity analyses for the criteria, and repeat the analyses with adjusted alternative and criteria weighting factors in order to expand the scope of the investigation.

6.0 Example: Analysis of Requirements for a Transportation Planning Software System

We experimented with the proposed techniques using a medium-size software system as the test case. In the following discussion, we modified some of the details and terminology due to proprietary concerns. This software system is currently being developed by the Carnegie Group, Inc., the employer of two of the authors. It is intended for transportation planning and will be used by a Government agency. Part of the requirements hierarchy is depicted in Table 4.

- Establish and maintain basic transportation data
    - Define and maintain transportation assets information
        - Port
        - Vehicles
        - Intermediate storage points
        - Regions
        - Preferred routes
    - Enter, display, modify, delete data
    - Archive/Unarchive data
- Create transportation solution
    - Obtain and manage transportation request
        - Manage cargo information
        - Manage movement request information
        - Manage handling information
    - Obtain and manage information on availability of vehicles
    - Obtain and manage information. on available storage/transfer assets
    - Generate transportation solution automatically
        - Enable human to set solution parameters
        - Generate additional missions for available vehicles
        - Assign cargo to vehicles
        - Satisfy capacity and routing constraints
        - Provide approximate optimality
    - Repair plan disturbances
        - Extract disturbances information
        - Evaluate effect of disturbances
        - Revise plan to resolve disturbances
    - Evaluate quality of plan
        - How many movement requests were satisfied
        - Delay time
        - Combined measure of merit
    - View/Edit the plan
        - View/Edit cargo itinerary
        - View/Edit mission route and timing
    - Manage up to ten alternative plans
- Disseminate and reconcile solution
    - Design a plan as the approved executing plan
    - Formulate mission requests
    - Forward mission requests to five executing organizations
    - Formulate vehicle manifests
    - Forward vehicle manifests to eleven executing organizations
    - Obtain objections from executing organizations
    - Formulate objections to plan disturbances
- Monitor the transportation process
    - Support cargo in-transit visibility
        - Obtain cargo actual itinerary
        - View/Edit cargo actual itinerary
        - Formulate disturbance and submit for replanning
    - Support visibility of missions
        - Obtain mission changes/cancellations
        - View/Edit mission changes/cancellations
        - Formulate disturbance and submit for replanning
    - Support visibility of port/storage facilities
        - Obtain changes in capacity/availability of facilities
        - View/Edit facilities changes
        - Formulate disturbance and submit for replanning
    - Present overall view of execution to human monitor

Table 4: Transportation problem requirements hierarchy.

Requirements rankings were performed by several engineers and managers. Some of them have had experience with the development of systems with similar requirements. The group followed the technique described in Section 2.

To address the concern about cost overlaps between requirements, we offered the analysts the following advice: select one of the requirements as the "base" that will absorb the shared cost. Record your assumption. The other requirement(s) can then be viewed as "add-ons" which take advantage of the cost already absorbed by the base requirement. Analysts found this approach fairly easy to follow.

An example matrix is shown in Table 5. The resulting measures of cost, benefit and risk are shown in Table 6. The members of the group made a number of observations:

- The approach was very simple to use. Even though only 20% of the group had had any previous exposure to AHP and the rest received only ten minutes training, the participants did not have any difficulties.

- Adequate time should be provided to fill in the matrices. A typical matrix of 5x5 requires about ten minutes.

- Several participants felt that they needed some sort of feedback, e.g., seeing the ranking of the requirements while they fill in a matrix. It would be useful for each participant to have software with such capability.

- Ranking requirements in terms of benefits was difficult and uncertain, especially for lower-level requirements, many of which do not contribute directly and independently to the satisfaction of user need. Perhaps ranking of requirements in terms of benefit should be limited only to high-level requirements which provide direct contributions to the end-user benefits.

- Ranking in terms of cost was the easiest. Risk fell somewhere in between cost and benefit.

- It is desirable to have three different sessions: one dedicated to evaluation of cost only, another for benefits and a third for risk. We performed these evaluations in one session and this resulted in risk and benefit measures being seriously influenced by the cost.

- Contrary to our original expectations "overlap" between the requirements has not been an issue.

	Establish and Maintain Basic Transportation Data	Create Transportation Solution	Disseminate and Reconcile Solution	Monitor the Transportation Process
Establish and Maintain Basic Transportation Data		c = 1/7 b = 1/9 r = 1/7	c = 1/3 b = 1/5 r = 1/5	c = 1/3 b = 1/3 r = 1/2
Create Transportation Solution	c = 7 b = 9 r = 7		c = 7 b = 5 r = 5	c = 7 b = 1 r = 7
Disseminate and Reconcile Solution	c = 3 b = 5 r = 5	c = 1/7 b = 1/5 r = 1/5		c = 3 b = 1 r = 1
Monitor the Transportation Process	c = 3 b = 3 r = 2	c = 1/7 b = 1 r = 1/7	c = 1/3 b = 1 r = 1	

Table 5: Pairwise relative contribution of requirements using cost (c), benefit (b), and risk (r) measures.

- Establish and maintain basic transportation data - c = .056 b = .054 r = .058 Comp. = 16.626
- Create transportation solution - c = .679 b = .503 r = .653 Comp. = 1.135
  - Obtain and manage transportation request - c = .049 b = .032 r = .057 Comp. = 11.457
    - Manage cargo information - c = .008 b = .007 r = .008 Comp. = 109.375
    - Manage movement request information - c = .022 b = .015 r = .017 Comp. = 40.107
    - Manage handling information - c = .019 b = .010 r = .012 Comp. = 43.860
  - Obtain and manage information on availability of vehicles - c = .022 b = .022 r = .023 Comp. = 43.478
  - Obtain and manage info. on available storage/transfer assets - c = .022 b = .030 r = .023 Comp. = 59.289
  - Generate transportation solution automatically - c = .306 b = .177 r = .210 Comp. = 2.754
    - Enable human to set solution parameters - c = .009 b = .004 r = .005 Comp. = 88.889
    - Generate additional missions for available vehicles - c = .021 b = .012 r = .015 Comp. = 38.095
    - Assign cargo to vehicles - c = .069 b = .035 r = .043 Comp. = 11.796
    - Satisfy capacity and routing constraints - c = .121 b = .075 r = .092 Comp. = 6.737
    - Provide approximate optimality - c = .087 b = .050 r = .056 Comp. = 10.623
  - Repair plan disturbances - c = .158 b = .138 r = .203 Comp. = 4.303
    - Extract disturbances information - c = .009 b = .008 r = .014 Comp. = 63.492
    - Evaluate effect of disturbances - c = .027 b = .024 r = .030 Comp. = 29.630
    - Revise plan to resolve disturbances - c = .122 b = .107 r = .159 Comp. = 5.516
  - Evaluate quality of plan - c = .040 b = .029 r = .032 Comp. = 22.656
  - View/Edit the plan - c = .020 b = .024 r = .024 Comp. = 50.000
  - Manage up to ten alternative plans - c = .060 b = .051 r = .034 Comp. = 25.000
- Disseminate and reconcile solution - c = .168 b = .188 r = .169 Comp. = 6.622
- Monitor the transportation process - c = .097 b = .256 r = .119 Comp. = 22.178

Table 6: Results of Evaluation - relative measures of cost (c), benefit (b), risk (r), and composite measure b/(c*r) for a part of the requirements hierarchy.

We then applied the cost estimating approach proposed in Section 3. The two most experienced members of the group were assigned to the task of cost estimating. They selected a subset of the leaf-level requirements using the following (admittedly ad hoc) approach:

- Prefer those requirements with which cost estimators are most familiar
- Prefer those requirements that can be associated with a well-defined software module(s) and a distinct scope of work
- Select about 10-20% of all requirements
- Select requirements from as many different branches of the hierarchy as possible; the intuition here is that broader distribution will reduce estimating errors
- Avoid requirements with either very high or very low relative cost; the intuition here is that outliers are likely to be less accurate.

Table 7 shows the cost estimates for selected requirements. Because the primary component of the cost in software systems is the development time, the estimates were performed in person-months. The numbers shown in the table have been scaled to notional "time units" in order to avoid release of proprietary information. Note that there is not a strong correlation between the cost measures obtained using the AHP procedure and the cost estimates for individual requirements. Our AHP approach indicated that relative measures of cost for requirements "Generate Additional Missions" and "Evaluate Quality of Plan" were .021 and .040 respectively. The detailed cost estimate produced quite the opposite picture: 5.1 and 2.0 time units respectively. However, we did not expect a very strong correlation and this is the reason for using not a single leaf entity but a set of leaf-level requirements for estimating purposes. It could be advisable at this point in the process to review and adjust the relative cost measures in order to arrive at a closer agreement between the two estimates, at least for the extreme cases like the one we mentioned above. In this particular experiment we did not take such a step. One may question which of the two estimates is more reliable and which one should be adjusted. The overall cost of the system was then estimated as the sum of the values in the third column of Table 7 & divided by the sum of the second column, i.e., 23.79/.180 = 132.17. This estimate did not contradict estimates obtained by other, more time-consuming cost estimating methods.

Requirement	**Relative Cost Measure**	**Cost Estimate (time units)**
Enter, modify basic data	0.016	4.1
Manage movement request info.	0.022	2.3
Info. on storage/tranfer assets	0.022	1.7
Set solution parameters	0.009	1.3
Generate additional missions	0.021	5.1
Extract disturbances information	0.009	2.1
Evaluate quality of plan	0.040	2.0
View/Edit the plan	0.020	3.3
Formulate mission requests	0.012	0.09
Obtain mission changes	0.009	1.8
TOTAL	0.180	23.79

Table 7: Estimating cost of the total system based on a subset of the requirements.

Two approaches to design and implementation of this system have been identified and proposed by engineering teams. One concept (named Concept-R) involved re-engineering of an existing system and heavy reuse of the existing software. The other concept (named Concept-N) involved designing a new system from the ground up with close attention to the specifics of its requirements. While very similar in terms of operational benefits, these two approaches presented a number of serious tradeoffs in costs and risks.

We evaluated both approaches using the technique discussed in Section 4. Table 8 shows a part of the requirements hierarchy and the assignment of measures to the two competing concepts. Notice that some requirements are satisfied in Concept-R with lower cost than in Concept-N, while others present the opposite impact. The computation of cost measures are shown in Table

8. The b/(c*r) measure for the Concept-R was higher and the cost measure was lower. This was consistent with the preference of engineering management for the Concept-R.

Requirement	**Relative Cost Measure**	**Concept R - Cost Relative to Requirement**	**Multiply by Req's Measure**	**Concept N Cost Relative to Requirement**	**Multiply by Req's Measure**
Enter, modify basic data	0.016	0.5	0.00800	0.5	0.00800
Manage movement request info.	0.022	0.66	0.01452	0.34	0.00748
Information on storage/transfer assets	0.022	0.6	0.01320	0.4	0.00880
Set solution parameters	0.009	0.5	0.00450	0.5	0.00450
Generate additional missions	0.021	0.34	0.00714	0.66	0.01386
Extract disturbances information	0.009	0.5	0.00450	0.5	0.00450
Evaluate quality of plan	0.040	0.5	0.02000	0.5	0.02000
View/Edit the plan	0.020	0.25	0.00500	0.75	0.01500
Formulate mission requests	0.012	0.25	0.00300	0.75	0.00900
Obtain mission changes	0.009	0.5	0.00450	0.5	0.00450
etc...	etc...	etc...	etc...	etc...	etc...
TOTAL	1.000		0.473		0.527

Table 8: Two system implementation concepts evaluated with respect to relative costs.

7.0 Conclusions

We proposed several related uses of the Analytical Hierarchy Process for quantitative analysis of requirement specifications organized in hierarchical structures. Our focus is on techniques that can be applied at different stages of the requirements definition process, including very early stages, and which require a very modest amount of effort.

A requirements analyst often faces a need to select among alternative decompositions of a high-level requirement (also called a need). We described a way to apply the AHP approach to ranking the alternative decomposition. An analyst can also use assistance in identifying a suitable set of criteria for such a ranking. We propose that in a computer-aided requirements engineering requirement with semi-formal representation of requirements, the program can identify and propose to the analyst a set of candidate criteria.

Comparative evaluation and tradeoffs between individual requirements is another common concern of a requirements analyst. An AHP-based technique proposed here provides a consistent and low-effort method of estimating the relative impact of each individual requirement on the system's overall cost, benefits and risks. These measures can be used to identify poorly formulated requirements, unnecessarily expensive requirements and suitable candidates for cost and risk tradeoffs.

A particularly interesting extension of this approach is a technique for estimating the total cost of the system proposed in this paper. While at this time we do not have conclusive experimental support for the validity and accuracy of this cost estimating technique, we see it as appealing to system engineering practitioners and expect to see further validation studies.

Finally, we discuss a technique for ranking competing system proposals or design concepts using the AHP-based evaluation of individual requirements.

8.0 Acknowledgments

This work was supported by the Human Systems Center (AFMC), Armstrong Laboratory, Logistics Research Division, Wright-Patterson AFB OH. Sumaria Systems, Inc. was the prime contractor of the project, Carnegie Group, Inc. was a subcontractor. The development of RAPID-WS, including its requirements analysis and prioritization capabilities, was guided by Ms. Janet L. Peasant of Air Force Armstrong Laboratory. The authors are indebted to Mr. Richard Nill of Carnegie Group who provided valuable insights, to Sumaria Systems and Carnegie Group engineers who designed and built the RAPID-WS software, and to Dr. Douglas Popken who suggested numerous improvements in this paper.

9.0 Bibliography

Air Force Instruction 10-601 (1994) Mission Needs and Operational Requirements, Guidance, and Procedures, HQ USAF/XOR, May 31.

Albrecht, A. J. and Gaffney Jr, J. (1983) Software function, source lines of code, and development effort prediction: a software science validation. IEEE Transactions on Software Engineering, SE-9, pp. 639-648.

Boehm, B. W. (1989) Software Risk Management, IEEE Tutorial.

Boehm, B. W. (1984) Software Engineering Economics, IEEE Transaction on Software Engineering, SE-10, pp. 10-21.

Booch, G. (1986) Object-Oriented Development, IEEE Transactions on Software Engineering, SE-12.

Hester, S.D., Parnas, D.L. and Utter, D. F. (1981) Using Documentation as a Software Design Medium, The Bell System Technical Journal, 60.

Jones, C. (1986) Programming Productivity, McGraw-Hill, New York.

Karni, R., Sanchez, P., and Tummala, R. (1990) A Comparative Study of Multi-Attribute Decision-Making Methodologies, Theory and Decision, Volume 29, pp. 203-222.

Keeney, R.L. and Raiffa, H. (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Wiley, New York.

Kott, A. and Peasant, J. (1995) Concurrent Engineering Research and Applications, Volume 3, n.2, pp. 93-106.

Korhonen, P., Moskowitz, H., and Wallenius, P. (1992) Muliple criteria decision support - A review, European Journal of Operational Research, Volume 63, pp. 361-365.

McDermid, J. A., (1993) Software Engineer's Reference Book, CRC Press.

Narasimhan, R. and Vickery, S. K. (1987) An Experimental Evaluation of Articulation of Preferences in Multiple Criterion Decision-Making (MCDM) Methods, Decision Sciences, Volume 19, pp. 880-888.

Popken, D. A. and Perkins, J. W. (1996) An Information Framework for Requirements Management, Proceedings of the 64th Military Operations Research Society Symposium, June.

Reitveld, P. and Ouwersloot, H. (1992) Ordinal Data in Multicriteria Decision Making, a Stochastic Dominance Approach to Siting Nuclear Power Plants, European Journal of Operational Research, Volume 56, pp. 249-262.

Roy, B. (1968) Classement et Choix en Presence de Points de Vue Multiples (la Methode Electre), Revue Francaise d'Informatique et de Recherche Operationalle 8, pp. 57-75.

Roy, B. and Bertier, P (1973) La Methode ELECTRE II, une Application au Media-planning, In OR 72, M. Ross (ed.), North-Holland, pp. 291-302.

Saaty, T.L. (1980) The Analytic Hierarchy Process, McGraw-Hill, New York.

Saaty, T.L. (1986) Axiomatic Foundation of the Analytic Hierarchy Process, Management Science 32, 7, pp. 841-855.

Saaty, T.L. (1987) Risk: Its Priority and Probability, the Analytic Hierarchy Process, Risk Analysis 7, 2.

Saaty, T.L. (1993) Multicriteria Decision Making: The Analytic Hierarchy Process, RWS Publications.

Saaty, T.L. (1996) The Analytic Network Process, RWS Publications.

Saaty, T.L. and Vargas, L.G. (1979) Estimating Technological Coefficients by the Analytic Hierarchy Process, Socio- Economic Planning Sciences 13, pp. 333-336.

Saaty, T.L and Vargas, L.G. (1982) The Logic of Priorities, Kluwer-Nijhoff Pub.

Saaty, T.L. and Takizawa, M. (1986) Dependence and Independence: From Linear Hierarchies to Nonlinear Networks, EJOR 26, pp. 229-237.

Schuijt, H. (1994) Comparison between ELECTRE III and AHP/REMBRANDT, Report 94-34, Technical Mathematics and Informatics, Delft University of Technology, The Netherlands.

Simon, H. A. (1962) The Architecture of Complexity, Proceedings of the American Philosophical Society, 106, pp. 467-482.

Vargas, L.G. (1994) Comparison of Three Multicriteria Decision Making Methodologies: The Analytic Hierarchy Process, Multiattribute Utility Theory and Outranking Methods, Proc. of the 3rd Int'l Symposium on the AHP, Washington, D.C., July.

Vessey, I. and Conger, S. A. (1994) Requirements Specification: Learning Object, Process, and Data Methodologies, Communications of the ACM, 37, pp. 102-113.

Yourdon, E. and Constantine, L. L. (1985) Structured Design: Fundamentals of a Discipline of Computer Program and System Design, Prentice-Hall.