An Approach for Dealing with Dynamic Multi-Attribute Decision Problems

Rachel Edith Baker

Submitted in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

University of York

Department of Computer Science

October 2005

Abstract

There will be a requirement for future systems to increasingly undertake self-management functions, one aspect of which is the dynamic resolution of what are, effectively, multi-attribute decision problems. Such problems involve acts of choice and selection based upon the consideration of several issues. Whilst there are many techniques to guide a decision maker to a satisfactory solution where problems arise once, and in a static context, these may not be readily suitable for dealing with dynamic problems encountered in a system's environment.

This thesis investigates the definition of multi-attribute decision problems within a real-time system. It proposes a methodology by which to structure the definition phase of the problem and to help in the identification of an appropriate evaluation methodology. Adopting a reasoned approach towards the definition of the problem also allows for the simultaneous description of adaptive behaviours by which to improve the robustness of the evaluation method without undermining the integrity of the results it produces.

Table of Contents

List of Figures

List of Charts

List of Tables

Acknowledgements

I would like to express my gratitude towards the following people for the assistance they have provided with respect to this research: Dr Neil Audsley (Supervisor), Professor Andy Wellings and Professor Mike Rodd (Internal and External Examiners respectively) together with all the members of the Real-Time Systems Group, particularly Mrs Sue Helliwell, for their encouragement over the years.

I also extend thanks to BAE Systems and the EPSRC for their generosity in providing the funding which made my attendance as a research student at the University of York possible.

Declaration

I declare that the work presented in this thesis is original work undertaken whilst registered for the degree of Doctor of Philosophy at the University of York between October 2001 and September 2005. Some of the ideas contained in this work featured in the following paper:

R E Baker and N C Audsley, "A Selection Mechanism Based upon the Explicit Expression of Preference", Workshop on Self-Managed Systems, SIGSOFT 2004, Newport Beach, CA, 31 October - 1 November 2004.

Chapter 1

The Advantages of Being Uncommitted

The advantages in being uncommitted to a particular course of action are that when the time comes to make a decision the most recent and relevant factors can be taken into account, leading to a better choice being made and, potentially, a greater level of satisfaction with the consequences of that choice. Likewise, the greater the number of factors which are considered, then the more thoroughly can a decision problem be examined. These two points form the fundamental argument behind enabling dynamic multi-attribute decision resolution in a system context. To do so would allow behaviours to be customized to the prevailing requirements and lead to improved levels of satisfaction for both processes and end-users. However, there are many things which, whilst they are theoretically desirable, are practically difficult to achieve and solving multi-attribute decision problems dynamically might prove to be one of these. The difficulty lies not so much with the production of a result, which might be readily generated, but in the production of one which is a valid representation of the underlying value structure and against a background which may allow both the value structure and the elements of the problem to vary as the problem is repeatedly encountered. Of course, it should not be forgotten that there are also disadvantages associated with a lack of commitment, principle of which is the increased uncertainty attached to outcomes.

This thesis explores issues relating to the deployment of dynamic multi-attribute evaluations in a computer based system and, by proposing that the subject be approached in a structured manner, aims to reduce the divergence between what are seen as desirable system properties, for example predictability and objectivity in behaviours, and their inverse realisations of unpredictability and subjectivity, the latter being traits which are more likely to be associated with dynamic decision scenarios. The term *dynamic* is used to indicate the on-line evaluation of decision problems which have been anticipated whilst the phrase *multi-attribute evaluation* indicates that it is necessary to consider more than one criterion in attempting to identify a solution to these problems.

1.1 Motivation

As in other walks of life, choice is seen as something of a mantra to success, and modern systems, whether embedded, distributed or desktop are no exceptions. The user is presented with a multiplicity of applications with which to perform the same task, networks to log on to and configurations to customize with their own preferences. Meanwhile, the system is faced with a host of processes to manage, hardware devices to access and network traffic to differentiate between. Decision scenarios can reflect situations of over-abundance where more elements are supplied than are required or, at the other extreme, they can reflect situations of insufficiency, for example where a failure has occurred and the system must take remedial steps to compensate for it. Such instances present the decision maker with the possibility of exercising his beliefs and resolving the problem dynamically.

The frequency, however, with which decision structures are encountered in the operation of a general system is somewhat at odds with that of the real world. This is due to both historic and technical factors: hardware limitations, in terms of size and speed, discouraged the addition of optional components; critical systems employing a hard real-time model discouraged the unpredictability of such approaches and best-effort systems were insufficiently advanced to offer much by way of extra services to warrant the user needing to make a choice. The situation over recent years has changed, though, and not only have systems become quicker, larger and more prolific, but users have also become more sophisticated and demanding. In addition to the general increase in awareness and knowledge, which has led to system expansion in all dimensions and prompted a perceived increase in the requirement for dynamic decision resolution, there are two other characteristics of system development which have had an influence on this topic. One is the increase in the complexity encountered both in system design and system operation, particularly in bespoke systems. This has raised the possibility of shifting decisions which were previously undertaken manually into the system to be effected at runtime, task scheduling being a prime example [Bur01]. The other is the increase in what are, ostensibly, quality-driven operations, primarily in the guise of Quality of Service (QoS) management [ISO98] but also in the more general stance of product enhancements and customized add-ons which give the edge in the consumer market. Interpreting the term ‘quality’ in the broadest sense it can be envisaged that, in conjunction with the user employing his personal value structure to define the appropriate form of problem resolution, dynamic decision scenarios will have to deal with the issue of user subjectivity, for example accounting for a user's preferences when undertaking the selection of a software tool for a particular job. This interface between human subjectivity and machine objectivity is one of the key aspects which needs effective management in future systems.

Listed below are illustrations of the types of system context where dynamic multi-attribute decision scenarios may have a role to play (note the predominance of recent research paradigms). To clarify, the characteristic nature of the decision problems being considered here is one where the definition of the problem and the value structure by which it can be satisfactorily concluded, have been provided by one or more human decision makers, but the actual evaluation and resolution is left to the system to undertake at runtime, possibly on a repeated basis.

Ubiquitous/Pervasive Systems: The ubiquitous computing environments, as described by Weiser [Wei91], relate to situations where computers are embedded in the environment in numbers and densities hitherto unseen, feeding back data on their surroundings in order that human and system actions can be modified accordingly. One potential feature of processes and hardware deployed in such systems is their *context awareness* which enables them to respond to, for example, the current geographical location, user's current activities, other currently supported devices and remaining energy reserves [Dey01][Mar00]. The direct interaction with environmental conditions provides an abundance of information which is unknown a priori, requiring on-line management, possibly by suitable middleware processes [Yau04] and creating a potential for numerous decision points to be engaged. Another consequence of this multiplicity of computing nodes entering into all areas of life, together with the switch to network based activities, is the overwhelming amount of data which many processes and users will find themselves having to deal with. Data filtering, aggregation, selection and storage are tasks which can involve high volumes of data and critical processes needing a timely response, thus choices need to be made about how to extract, refine and deliver the specific data required [Coh01].

Mobile Systems: This general description covers a range of system types. At one extreme there are the mobile ad hoc networks (MANETs) of unmanned vehicles undertaking monitoring and surveillance tasks, self-organizing via the use of co-operative network management processes and running in-network data processing so as to reduce the subsequent network load [Dur01]. At the other end of the scale there is an individual user wandering around with his laptop, phone or personal organizer, logging onto wireless networks, as and when, to download data and update software as he feels necessary [Nob97][Nob00][Kat94]. Mobility and miniaturization have introduced new problems of size and energy constraints which necessitate trade-offs being made during the course of operations [Fli99][Til02].

QoS-Driven Management Schemes: The rise of networked systems, resulting in applications communicating over wired or wireless links has stimulated a concern about the fluctuations in the level of performance which can be experienced, particularly owing to the unpredictability of network traffic and the occurrence of bottlenecks within communication links which give rise to packet delays and losses. Additionally, interest has focussed upon high bandwidth consuming multi-media applications, for example streaming video applications, which can be described as having soft real-time characteristics, that is to say they may exhibit a certain flexibility in the observation of their timing requirements. This flexibility is seldom sufficient, however, to compensate for the best-effort nature of heterogeneous networks without additional QoS support. Such support motivates application and system adaptation together with mechanisms which enable traffic differentiation [Li99][Wu01][Bra97][Van99][Wan99][Fis98][Pol02]. The relevance to the topic of dynamic decision making is that such operations involve many decisions being made informally as to which elements (process parameter, packet characteristic, service, route etc.) receive favourable treatment and which do not. A particularly problematic area, in this respect, is deciding what information to provide to users so that they are able to make appropriate decisions. Additionally, issues relating to how to set about capturing user-level requirements effectively, and in such a way that they can be mapped down to the appropriate lower level of system operation, need to be addressed.

Autonomic Computing: The final area where dynamic decision scenarios might expect to be encountered is in that of autonomic computing [Kep03], a term which describes the concept of self-managed systems. IBM's interpretation of such a system identifies three hierarchical decision management layers encapsulating resources, groups of resources and the specific business context [IBM05], each designed to delineate the scope of the management functions' control loops. In their vision, autonomy decomposes into four areas of activities: self-management, self-configuration, self-optimization and self-protection. This will (theoretically) reduce the workload of system developers and system managers. In order to perform all these introspective tasks, however, the system must not only have the capacity to detect and monitor problems as they arise, but it must also have access to unambiguous data reflecting the value structure to be employed, and rules by which to apply it, so as to bring about an informed response to such problems. Self-adapting systems which can dynamically reconfigure themselves, either in response to a change in mission or as a consequence of a fault arising, will again require decision making capabilities [Ore99][Gar03].

It can be seen that there are plenty of opportunities for applying dynamic decision resolution and it is inferred that there are plenty of advantages in doing so. Assuming the concurrent development of monitoring systems and procedures by which to police interactions and safeguard correctness at all times, then there would seem to be little to prevent a more frequent adoption of such scenarios. However, such optimism overlooks some salient features associated with the process of decision resolution:

- It is generally understood that the use of decision theoretic methods to address decision problems does not result in the *correct* answer being identified, only the most desirable under the decision maker's value structure at that moment in time. This has implications for validation as correctness can be validated precisely whereas desirability is more difficult to prove, especially when it is based upon the perceived attractiveness of certain traits, encapsulated as attributes, some of which might be traded-off against each other.

- Employing traditional decision theoretic techniques may not, necessarily, be the appropriate choice in a dynamic context as these are designed to address a static and single instance of a decision problem and thus can fix the decision maker's preferences to a single point in time. Dynamic multi-attribute decision resolution brings with it an implication of repeated occurrences of the decision problem and, therefore, the decision maker's value structure should ideally also be allowed to vary over time as conditions change.

- For existing non-dynamic multi-attribute decision problems, the general decision support tools available may not be without criticism [Kuj03] or found inappropriate for a particular context [Ncu02].

If static decision problems, albeit of a different scale of activity, can generate a certain confusion and complexity then surely some thought must be given to the use of the same decision support methods before they are applied to the resolution of dynamic scenarios? There are many aspects to be considered before a succinct and satisfactory model for multi-attribute decision resolution in a system context is arrived at. One particular problem, for example, is how to undertake the elicitation of value structures from human decision makers in an adequate manner, including mechanisms for guiding trade-off choices. A starting point, it is suggested, is to view the problem scenario in its entirety and approach its definition in a structured fashion.

Of course, an immediate question raised when introducing the idea of employing dynamic decision resolution, and indeed adaptable behaviours in general, particularly in time-aware environments, is that they inherently introduce uncertainty into operations. As Thiele notes,

> *"Higher degrees of predictability can be achieved if static decisions instead of dynamic decisions are being used."* [Thi04]

However, given that one of the major threats to predictability, particularly in a distributed context, is the disruption arising from the system's inability to compensate for unknown elements then it may be that there is little option but to investigate and deal with such elements on-the-fly.

1.2 Thesis Proposition

This thesis is concerned with ensuring that the dynamic evaluation of problems of choice is undertaken in a principled manner. There are an increasing number of examples where it might be seen as advantageous to performance if system components were enabled to undertake the dynamic resolution of multi-attribute decision problems, rather than restrict behaviours to follow a statically defined path. However, in a real system context there are numerous obstacles associated with both the definition and execution of such activities and it is important to understand the nature and requirements of the problem involved in order that the chosen form is sufficient for the intended purposes. Such obstacles need to be considered and, therefore, this thesis proposes:

> *Within the context of a real-time system, approaching the definition of a dynamic multi-attribute decision problem in a structured manner can reduce the potential for inconsistencies emerging during the definition phase and allow any flexibility inherent within the problem's definition to be exploited concomitant with the design's progression and in such a way that it enables the future evaluation process to exhibit adaptive behaviours which will not undermine the integrity of the results it produces.*

That is, not only would dynamic evaluation scenarios have a role to play in supporting adaptation processes, they themselves may also exhibit adaptive capabilities. In order to do so, however, the subject must be approached in an integrated fashion and by practitioners who have a full understanding of the implications of any actions taken. Of course, in order to facilitate such developments it is beneficial to provide some guidance and mechanisms by which to shape activities which this work subsequently aims to do.

1.3 Structure of Document

This document assumes the following structure:

Chapter 1: Introduction and motivation underlying this work.

Chapter 2: Discusses a selection of literature thought relevant to the topic of dynamic multi-attribute decision problems, with the objective of placing the work in context.

Chapter 3: Introduces the generic model together with terminology relevant to the discussion.

Chapter 4: Undertakes to demonstrate a comparison of three possible evaluation methodologies in order to show how the choice of method influences the practicalities of operation and the results produced.

Chapter 5: Describes the analytical framework by which to approach the problem of realising the definition of a multi-attribute evaluation in a structured fashion.

Chapter 6: Provides an interpretation as to how the previous framework might be applied to a particular sensor-based problem.

Chapter 7: Examines the employment of adaptive evaluations on-line, using the techniques introduced in Chapter 5.

Chapter 8: Provides a conclusion to the previous sections and notes the contribution made.

Chapter 2

Related Literature

The theme of this work is the considered employment of dynamic multi-attribute evaluation methods. The focus is on the analysis of the problem and approaches by which the suitability of a chosen evaluation methodology, leading to the production of a 'value' rather than on the use of the value itself, can be determined. The consequence of performing an evaluation of several alternatives is assumed to be a single value assigned to each alternative by which its desirability with respect to the problem being investigated can be interpreted. It is the processes leading up to the production of this value which are of interest. Without being prescriptive about the form or context of such evaluations, it is apparent that this is a multi-faceted problem which cuts across several research domains. This chapter introduces work which is deemed of relevance to this subject and which serves to illustrate the types of scenarios for which a wholly statically defined response to the problems encountered dynamically would be inadequate.

The objective in using dynamic evaluation is to delay the point at which behaviours must be fixed in order that they can take account of more recent updates in internal and external conditions. In short, it makes the system more flexible which, in turn, can enhance other aspects of performance, for example improving fault tolerance. However, flexibility requires careful management if it is not to undermine the integrity and stability of operations. It also requires that the theoretical models and mechanisms by which it is realised be understood otherwise the validation process risks becoming obfuscated.

The generic decision problem at the centre of this discussion has a solution which is arrived at by the appraisal of two or more attributes, representing the state of alternatives, and where the consequence is known, i.e. there is no uncertainty to be directly accounted for. This simplifies the decision problem as it removes the requirement to handle probabilities and consider different expectations as to possible outcomes.

The material presented here is broadly structured to consider the reasoning which lies behind the use of such customized values and then to show examples of how values of this nature are employed in various system contexts. In short, it covers the *when, where,* and *how* of dynamic evaluation in contrast with the *what* and *why* which were touched upon in Chapter 1. Particular reference will be made to the world of real-time scheduling, in part because it is a familiar area, but more significantly because it represents a well-established example of an evaluation scenario with static, dynamic and value-based variants. Three particular strands of interest in this discussion are: the meaningful use of multi-criteria evaluation in a dynamic context; enabling such an evaluation to be adaptive; and the more explicit representation of issues of preferences. These are only some of the many factors involved, though, and amongst the others which are not dealt with here, there is, for example, the question of how suitable are extant decision theoretic methods to the resolution of dynamic problems. Section 2.1 discusses examples from the literature where thought has been given to the processes leading up to value production while Section 2.2 presents a sample of work where such values are made use of during operations.

2.1 Reasoned Value Production

Given the previous introductory statements the subject matter of this thesis might be interpreted as being the "reasoning about evaluation". As such it is similar in intent to work undertaken in the Artificial Intelligence community on *meta-reasoning*, or the "reasoning about reasoning", discussed further below. Additionally, it continues the theme established by Prasad [Pra98], also discussed below, in that it is concerned with the inherent validity of values when used. If values misrepresent the situations they are intended to reflect then why go to the trouble of producing them? Decision theoretic approaches feature in both these areas of work, as elsewhere.

2.1.1 Reasoning about the Choice of Software Process

Computational results are produced at a cost which can be measured in terms of time and system resources, raising the possibility of being able to trade-off quality against both time and resources in order to achieve a partial result in situations where constraints on performance are present. These possibilities were the focus of much attention from the mid-1980s onwards, particularly from research in the area of intelligent systems, where the intention was to provide models which would allow such trade-offs to be described and effected both on- and off-line. In attempting to achieve this objective two inter-related features were employed. Firstly, the use of computational time to reason about the form which any particular decision resolution should take. Secondly, the assumption that for many decision problems partial results could be produced which were still of benefit to the system by undertaking less than optimal evaluation. The generic decision problems being addressed were those where the decision maker was an agent faced with continual problem solving scenarios in situations of uncertainty. Consequently, solutions frequently made use of decision theoretic approaches, particularly that of Expected Utility as developed by von Neumann and Morgenstern [Neu47].

Russell and Wefald employed the term "metareasoning" to describe the process of reasoning about possible changes to the computational state of an agent [Rus89]. Alternatively, the activity is known as "deliberation scheduling" [Bod94], which effectively means, according to Dean,

> *"running a decision procedure whose purpose it is to determine what other decision procedures should run and when"* [Dea91]

Dean, in summarising Russell and Wefald's approach, distinguishes two sets of actions, *inferential* actions which affect the internal state of an agent only and consume computational resources, and *physical actions* which change the external state of the agent and may also require computational resources [Dea91]. In a real decision scenario, if too long is spent on the inferential state then the choice of physical action which emerges may no longer be the valid one to take as conditions may have changed. The basic objective of the agent is to maximise its utility $U(a)$ where A is the set of states of the world and $a \in A$. The agent has access to a set of computational actions $\{S_i\}$ which affect its internal state only but which may lead to a revision of the decision as to which is the best action to take next. Deciding what further deliberation to take is based upon establishing the net value of computation [Dea91],

$$V(S_i) = U([S_i]) - U([\alpha])$$

where $V(S_i)$ is the value of undertaking action S_i and $U([S_i])$ and $U([\alpha])$ are the utilities of the external states which result from undertaking actions S_i and the default action, α, respectively. If S_i reflects a partial computation then it need not lead to a commitment to undertake an external action whereas a complete computation would [Rus89]. The agent must, therefore, estimate the utilities involved and employ meta-reasoning to decide whether further evaluation is required, on the basis of net value, before taking an action.

The computational element of a problem may be characterised as being "quantised chunks" [Bod94], as in the work of Etzioni [Etz89] where the agent, given a set of goals and set of methods by which to achieve them, has the problem of identifying the optimal sequence of methods to deploy. Alternatively, it can be viewed as a continuous function of the time spent in evaluation as with the *anytime algorithms* proposed by Dean and Boddy [Dea88] or the *flexible computations* of Horvitz [Hor87]. Here, the object related value of a computation is time dependent and, if interrupted, the computation will return a value appropriate to the length of execution so far. Computational value, for example a measure of the level of precision achieved by the computation, must be distinguished from the overall value of the system's response to a given state, which will take in the utility delivered as a consequence of the state together with the costs involved in achieving it. This approach ideally suites algorithms capable of delivering a computational value which monotonically increases with time. Boddy notes that it can simplify problems where the output of one deliberation acts as input to another [Bod94] and, in answer to the question of how useful and generally available are anytime algorithms likely to be, they list some examples of the type of computation they have in mind, e.g. heuristic search and Monte Carlo algorithms.

From one perspective, the work on flexible computation can be interpreted as the study of algorithm selection techniques and in [Hor88] the author considers the multi-attribute nature of such problems when used to define the utility of partial results. He notes that a focus on the purely temporal characteristics of partial evaluations may ignore the significance to an agent of resource availability and resource costs. There may be many "dimensions of value", Horvitz's term, which can be ascertained through querying users. For example, if considering how to distinguish between the different partial results produced by various sorting strategies, then relevant criteria may include disorder, high/low-end completion and bounded disorder, all aspects relating to the relative positioning of, and distance between, fields within the sort. Other approaches to the method selection problem include that of Fink [Fin98] who describes a method for the automatic selection of a problem-solving method, prior to the analysis of the problem, and which differentiates between methods on the basis of a statistical analysis of their past performance. There is also the work of Lagoudakis and Littman [Lag00], who are interested in selecting the most appropriate method during the analysis of the problem, using a technique based upon a Markov decision process.

Of course evaluation and decision making activities, outside of computerised scenarios, may be equally subject to time constraints and work done in other areas might provide useful input to any intended software based schemes. For example, the effects of time pressures and the level of information available upon the quality of decision making has been the subject of study in areas of psychology, operational research and decision theory [Ahi98]. To illustrate, in [Pay96] the authors compare several decision procedures with respect to their performance, together with the decision maker's behaviour, when operating under the stress of working to deadlines. The results of such explorations can be of help in designing user-centric activities and also in providing the rationale to guide how software based decision support methods are implemented.

Finally, when discussing the question of what choice of software process, the subject matter can be thought of as relating to software components rather than individual algorithms. Component based architectures motivate such recent technologies as JavaBeans [Bea] and .Net [Net] with their emphasis on re-use and plug-and-play development. A more generalised interpretation, however, sees components as being commercial off-the-shelf (COTS) products, whose use reduces the necessity and time associated with undertaking custom development. Both scenarios introduce a fundamental decision problem given the potential for a multiplicity of functionally similar units being available for consideration in any one situation. Such a problem may be framed as a dynamic one, where the difficulty is one of deciding how to optimize performance on-line by selecting the most appropriate component [Yel03][Oli02] or, more commonly, it can be perceived as a static design-time problem [Ncu02].

The latter situation is the focus of interest in the work of Kontio [Kon95a][Kon95b] who is concerned with ensuring that the problem of selecting a reusable component from several alternatives is performed in a systematic fashion. The author describes the OTSO method which enables the definition of evaluation criteria to be undertaken consistently by referencing a common set of factors of relevance to the issue of re-use, such as application requirements and project objectives. Then, by employing a common reference point, that is a baseline representing the minimum level of properties which an alternative must exhibit, comparisons on the basis of a cost/benefit analysis can be effected. The OTSO method requires a multiple criteria decision technique by which to unite the individual criteria and the author opted for the use of the Analytical Hierarchy Process (AHP) [Saa92]. Interestingly, a comparison was made between the use of the AHP and of a weighted scoring table (WST), a common choice in such scenarios, whereby the authors demonstrated how the choice of evaluation methodology can have an influence upon the nature of the decision maker's preferential expression. The participants in the selection process found that the behaviours of the latter were less readily understood and less conducive to the production of well-balanced weighting constants.

Kontio's approach is the closest example found in the literature to the work described in this document. Its intent and rationale are similar, namely that multi-criteria problems require a methodical treatment if their complexity is to become routinely manageable within a design environment. It differs, however, in that it is concerned with a specific single decision instance and does not consider the more general nature of the selection problem. Also, it can be criticised for placing heavy reliance upon the use of AHP, via the Expert Choice tool [Exp] which employs this multi-criteria decision aid (MCDA). Whilst it was acknowledged that the AHP method generated a larger workload, for example through requiring a greater number of comparisons of alternative criteria, in the opinion of the participants the method returned more information, such as rankings of criteria, which they found useful in guiding their judgement. An alternative way of interpreting this observation is that because the method, in its commercial incarnation, presented several ways of 'playing around' with the data, the final definition of the evaluation problem risks becoming somewhat distanced from the original set of preferences which the decision makers held at the outset of the process. There are two implications of this: firstly, that the more features which a decision tool supports then the more acts of refinement of the model might be expected, which can be interpreted as both a good and bad thing, and thus the choice of tool directly influences the results; secondly, as has been mentioned elsewhere, preferences, particularly as they relate to these types of industrial/engineering problems, will seldom be a constant set of values. The more time spent investigating a problem then the more opportunity there is for the decision maker's preferences to vary as he obtains a deeper understanding of the problem itself. A certain degree of revision may be beneficial, for example if one takes as a starting point the introduction of a new problem, then the initial set of preferential opinions of the decision maker may be ill-defined and, as yet, uncertain, so a degree of further investigation and consideration will lead to these opinions becoming better structured and thus produce an improved definition of the problem. However, it is suggested that there is a certain point in the whole process beyond which, whilst the preferences may continue to change, the overall problem definition is not actually improved to any substantial degree. Instead, the decision makers, if left unchecked, may cycle through a range of scenarios, each being perceived as a marginal improvement on the last, without bringing any great gain to the operation as a whole.

It can also be observed that, as a consequence of running a trial with two evaluation methods, presented with the problem of selecting a browser, the authors found that the two sets of results differed in their ordering of alternatives. They expressed some surprise at this outcome though, in fact, this might have been anticipated to a certain degree as, owing to the nature of the trial, the decision makers may not have expressed precisely the same belief structure between the two cases (they were separated by some two weeks' period), also the precision of the evaluations may not have been the same and there is the possibility of the AHP method not being entirely consistent in its behaviour, as is mentioned elsewhere. All these points only serve to show that it is no easy task to prove that the numerical value, which purports to be a direct interpretation of the decision maker's preference structure, is an accurate interpretation of such beliefs.

2.1.2 Meaningful Values

Whilst much has been made in the literature about the benefits to be obtained by the promotion of value-driven systems there is seldom any mention about ensuring such values are derived in a consistent and meaningful way. An exception, in this respect, is the work of Prasad who focuses on this issue and raises awareness of the fact that further work is needed on the related topics of how to assign values and how to use them [Pra98][Bur00]. The author's concern is to promote the formalisation of value usage sufficient to allow the approach to be employed in critical systems where dependability issues are paramount. In addition to the exploration of the representational aspects of value derivation, placed in the context of measurement theory, and the use of decision theoretic techniques by which to approach decision problems, the author also considers the production of values intended to be employed in value-based scheduling schemes. A methodology by which to decompose the problem is proposed, which tackles the question of potential runtime complexity by reducing the amount of dynamism to be considered on-line. Using an example of autonomous vehicle control, the set of alternative services, e.g. braking systems, are characterised by a single static attribute only, that of importance, whilst the range of system states which might be encountered are categorised as *modes* according to the states of the environmental conditions, computational load and application software. By employing the assumptions that within a mode, a fixed set of services are defined, and that the value of alternatives are constant then the runtime decision problem becomes one of identifying the appropriate conditions and settling upon an admission/rejection protocol by which to utilise the value.

The actual value itself, i.e. that subjective parameter which characterises each individual service alternative, is static in nature and, consequently, can be ascertained off-line. This is achieved through the exercising of expert knowledge, whereby a system designer uses his preference to undertake pairwise comparison of all services on the basis of their relative importance under a particular mode. In order to facilitate this process a decision tool based upon the AHP was employed to produce the composite scores which were than directly used as alternatives' values. One of the advantages cited for the use of such theoretical approaches is that it reduces the possibility of inconsistencies in the decision making process. By only considering optional services the complexity of the problem was reduced further (essential services being guaranteed to run anyway). The intention is that, in order to enhance the flexibility and robustness of the target system, at runtime optional services would be admitted to the system on the basis of the value they achieve. The author also notes that a value need not necessarily be used in its pure form but is likely to be subject to statistical treatment, particularly when playing a role in the admission process [Pra03].

In the scenario discussed above, values are proposed as a means to enable adaptive scheduling schemes [Bur98]. Whilst only a single criterion is actually examined on-line, the general intention is to ultimately consider multiple attributes and so, for example, in replacing a fixed-priority scheduling parameter, the value would assume a greater role in encapsulating a variety of issues than the parameter it is replacing. Multiplicity, however, must be scoped somehow to ensure that complexity is restrained to some degree and to avoid unnecessarily incorporating attributes into evaluation which are better dealt with elsewhere. As demonstrated, the example is effectively akin to having undertaken an off-line decision as to which services to favour in a mode change. It does, however, highlight some likely difficulties associated with employing dynamic multi-attribute evaluation, not least in performing the relative appraisal of several criteria.

In reducing the need for the on-line measurement of alternatives' attributes it has introduced the potential for other inconsistencies which must be safeguarded against, namely ensuring that the correct environmental condition is identified and the appropriate static value mapped to it. Also, the applied usage of the values in the example is to enable the scheduling of optional services/tasks according to their value-density figure [Dav95]. It is proposed that the optional services are rejected on the basis of an utilisation-based feasibility test with those that remain then being scheduled on the basis of their value-density. Although this case is only meant as an illustration it suggests that, depending upon the periodicity of the mandatory portion of the task set, it is possible that there may never be a sufficiency of spare capacity to run some of the optional tasks with lower threshold values, given the wide range over which the set of their values is expressed, i.e. [0.1, 438]. This is a consequence of the original values themselves, as produced by the AHP process, being wide-ranging and is a resolvable issue. If optional tasks are to be implemented as ‘helper’ functions to periodic tasks, then it is reasonable to expect that they are able to run at least once during the course of operations.

Another point to be made of a general nature relates to the use of tool support. The complexity of systems means that progress in design cannot be made without reliance upon off-the-peg tools. As in any situation caution must be exercised towards their reliability or, as in this case, towards the theoretical soundness of the underlying model. The AHP approach has been subject to some criticism with regard to certain inconsistencies in behaviour which can give rise to rank reversal [Tri95]. This raises a fundamental problem with the use of values, for example as in this particular scenario where the value does not change once assigned and scheduling feasibility is checked separately, namely if the actual values are a misrepresentation would the decision maker be able to detect such an error? If only a small number of alternatives and attributes are involved then it is reasonable to suppose that errors might be noticed as he is still likely to be able to relate them to his intuitive opinions. However, as the number of components involved increases, it may be more unlikely that the decision maker's ability to differentiate between alternatives is sufficiently keen enough to notice and his reliance on decision tools much greater. Collectively, these issues indicate that whilst some of the potential difficulties which might be expected to arise in the use of value-driven operations may be addressed they, in turn, may only give rise to the emergence of others.

2.1.3 The Direct Employment of Preferences

If multi-attribute values are to be employed within system operations, that is where one value summarises the state of more than one attribute, then whether dynamically derived or not, there is a need to differentiate between the relative significance of attributes unless, that is, they can all be guaranteed to be of equal significance in every instance. Two means to achieving this differentiation are for the decision maker to exercise his preferences over the set of attributes and for these to be realised as value trade-offs within the numerical representation applied. Examples which make explicit reference to expressions of preference can be found in the general literature but, as with the notion of utility which is discussed later, the term is liable to be used in a somewhat ad hoc fashion and without the deeper implications of its application being considered. Effecting a numeric representation of such preferences in a systematic way remains something of a complex issue, not least because, by and large, preferences may not be static entities.

Whether explicitly mentioned, or not, preferences underlie all aspects of system design and implementation. However, there is one area where they can be seen to be of immediate relevance, namely in undertaking customisation of services in response to user requirements. One illustration of such an approach can be found in the work of Chalmers which tackles the problem of implementing adaptive mobile applications [Cha02]. Adaptation is required, which the author refers to as "context mediation", to account for both users' requirements and the prevailing system context. The scheme described is demonstrated by way of an adaptive map application in which the user must specify two separate strands of preferences, firstly for the relative desirability of element types, e.g. road, river and the like, and, secondly, for how they perceive the quality of the presented data to vary according to the state of the attributes which characterise it. The former preference is reflected in a weighting function which can take several parameters as inputs and applies a weight to a particular element, for example the weight accrued by a motorway, which is a specialization of type road, might be dependent upon its sub-type and distance. The latter preference is summarised by a utility function which is applied to an attribute of the type of data, so the scale of a map, for example, which will affect the representation of the road elements, will have a perceived utility to the user which is dependent upon what their intended usage of the data is to be. The overall utility is then achieved via the combined product of the utilities of individual attributes. These details must be provided by the user who creates several specifications reflecting different contexts, devices etc., and are stored in XML format as meta-data descriptions to be queried as required. So, for example, if the user has specified a constraint on the time to download an area of map and the connection restricts the amount of data which can be delivered in that time then map components are selected for retrieval on the basis of the utilities given in the meta-data.

Leaving aside the issue of handling specification interactions, the previous description prompts two observations. The use of weighted utilities appears to be somewhat informal and implicitly incurs the possibility of attributes and data types being capable of being traded-off against each other. In some specifications the combination of values may be such that, although the user has stated this preference, he does not in reality mean it, for example if a user is driving he might express a preference for small roads and rivers as he likes to drive and look at ducks. If an area has a high proportion of rivers then these might take precedence over a single road but this information will be of no practical benefit as he will be unable to get there. Such idiosyncrasies can be safeguarded against by use of formalized proven models, suitable for the role intended, or they can at least be anticipated by vigorous testing procedures.

Additionally, it is readily apparent that user specification is not necessarily a straightforward process and would require a carefully defined interface if it is to be effected satisfactorily. In [Hal04], which considers the same scenario of resource constrained mobile devices and presents a self-adapting middleware framework, the authors note that defining the property characteristics and utility functions, which represent the user's perceived utility, is not without difficulty and that this aspect of elicitation and representation may require further expertise. Whilst the argument can be made for the advantages arising from employing value-driven operations, where the value is predicated on users' expressions of preference, there is a balance to be struck between allowing too little user involvement and burdening them with a requirement for too much. It is a fine line between enhancing user experience and overwhelming them with meaningless obligations.

The previous point is part of the motivation behind the work discussed in [Pol04] which deals with the same area of user specified adaptive systems. The authors are partly motivated by the desire to promote the automation of such services and partly to reduce the complexity associated with effecting reconfiguration. As they note, they focus on formalisms by which to represent user preferences within system operations rather than on exploring how such preferences should be extracted in the first instance. Users stipulate a weighted utility for each QoS dimension of a service, where the weight indicates how important that dimension is to them, examples of dimensions for a video application being the frame update rate, frame size and audio quality. They also express a similar weighted utility with respect to the specific supplier of a service, e.g. the particular type of text editor, though this is moderated by the inclusion of a penalty charge which represents the costs incurred in switching suppliers. The overall utility associated with a particular service is then a consequence of the product of the sum of its individual dimensional utilities and the supplier preference utility. Application profiles represent the resource requirements necessary to fulfil different levels of performance. The authors also acknowledge the theoretical requirement for preferential independence in specifying the model but feel satisfied that, even if these do not completely hold in practice, behaviour is still acceptable.

It is necessary to quantify expressions of preference in some way if they are to be of practical benefit within a system context but this may not be a simple matter. Some of the pitfalls of attempting to 'score' expressions of preference are noted in [Duj96] which describes a preference-driven decision method to facilitate the evaluation and selection of system components during the design stage. The proposed method takes the values assigned to individual criteria, expressed as a percentage of the maximum score achievable, and adopts a logical approach in combining these hierarchically into compound preference values. If a generalised scoring approach of a weighted additive form were to be employed, then, as the author points out, it would not be able to represent the mandatory nature of an individual attribute, as this would not yield zero in the cumulative score, and the difference between the magnitude of attributes' contributions would effectively place an upper limit on the number of attributes it was worthwhile addressing. Various other ways of combining multiple sources of data are reviewed in [Dan01] which discusses this issue of trade-off analysis, again from the perspective of the design process.

2.2 The Employment of Values

An overview of the infiltration of value-based operations into real-time systems is presented in [Wel01]. Here, in addition to motivating the case for the extended use of 'benefit', which is their preferred term to that of ‘value’, the authors also identify different types of value production according to the level of abstraction addressed. At the low-level is the use of benefit as a function of time based around the iteration of periodic processes, then comes benefit as a function of time and resource allocation, as seen in many QoS-driven resource management schemes, up to benefit as a function of both resource allocation and the output quality of the process, the latter case reflecting changes in the levels at which applications execute. They also note that, in assessing benefit, there are different levels of perspective to be considered, which they detail as thread, application, system and user. These will not necessarily be in accordance, particularly if there is more than one user. For example, the system's benefit may be maximised by running a certain selection of applications, whereas the user may consider their benefit maximised by running a different set. Re-formulated in terms of a decision resolution scenario, this description exactly captures one of the dilemmas involved in promoting the increased use of dynamic evaluation, namely identifying the correct level of decision maker and managing the interactions between decision makers, particularly where the system exhibits several competing goals. Such unstructured multiplicity may prove to be one of the delimiting factors on the uptake of value-based schemes.

Scenarios where dynamic values are employed include those dealing with the ‘unknown’, an excess of objects, an abundance of choice, QoS issues and the dynamic capture of user requirements leading to customisation of activities. In a system context these uses can be mapped to just two primary, but not necessarily exclusive, objectives which are the dynamic re-allocation of system resources and the expression of user preferences.

2.2.1 Value-Based Scheduling

The foundation of value-based scheduling, in the context of real-time systems, lies with the work of Jensen et al [Jen85]. Starting from the proposition that the most distinctive characteristic of such systems is,

> *"the concept that completion of a process ..... has a value to the system which can be expressed as a function of time"* [Jen85]

they introduced a time-driven scheduling model in which the scheduling parameter is derived from the application of time-value functions (TVFs). Over the years the terminology has fluctuated between ‘time’, ‘benefit’ and ‘utility’ but all descriptions refer to the same functional form. A task is characterised by a value function which returns the value to the system of completing the task at time *t*. The general objective in such scenarios is to maximise the accrued value to the system.

The advantage, as Locke notes, is that the time-value approach allows tasks with both hard and soft deadlines to be accounted for in the same model with the semantics of the latter being precisely specified. Tasks have a value which will vary with progress made from their release, t_0, until their deadline, D_i, is reached at which point the value becomes zero and the uncompleted task is removed. A critical time, t_c, such that $t_0 < t_c \leq D_i$ can indicate a significant change in utility, with the deadline being an extreme form of the critical time. This is in contrast with the use of a fixed-priority scheduling algorithm where all tasks are deemed to be of no value if they exceed their deadlines even though, in reality, certain tasks with soft deadlines may still be able to contribute some benefit to the system after they have overrun. In the original work the authors noted that the value functions could be provided by either the process implementer or the system architect, and assumed that,

> *"all sources of value information are reflected in the value functions used"* [Jen85]

with no further description of the derivation process being given.

Time-value functions can take on arbitrary shapes but are commonly assumed to be *unimodal* and *non-increasing,* that is they have a single optimal completion time and their utility cannot increase once it has decreased. Example functions are shown in Figure 2.1 indicating how utility can vary as time progresses. In the original work the time-value approach was used to evaluate several well-known scheduling schemes and to describe two new algorithms, *BEValue1* which exhibited greedy behaviour and scheduled tasks on the basis of largest value-density first, given as *value/computation time*, and *BEValue2,* which effectively operated on an earliest deadline first approach whilst the probability of overload was below a stated threshold and then eliminated accepted tasks on the basis of lowest value density until the probability of overload returned to an acceptable level. The demonstrated benefit of this second approach led the authors to conclude that employing values in overload scenarios would be particularly advantageous as it would allow for a level of predictability in behaviours which is otherwise absent. This theme has since been taken up in several other studies and will be returned to later.

Subsequent to the original work, a case study demonstrating the application of time-value scheduling to an airborne tracking system was presented [Cla99]. In this, the process of appraising the worth of activities was described further and serves to illustrate the potentially protracted and ad hoc nature of such design phase actions. The objective was to enable a

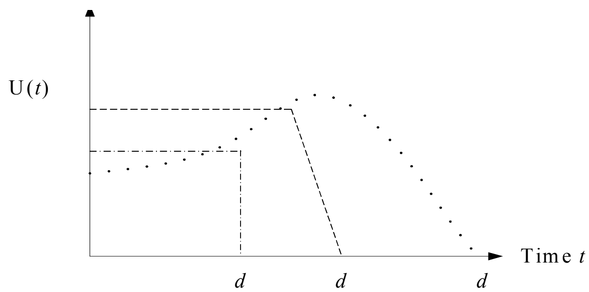


Figure 2.1: Examples of Time-Value Functions

tracker system to adaptively select the most important tracks to process from incoming sensor reports since the non-modified form had a maximum handling capacity which meant overloads could arise and, owing to the mechanics of sensor report retrieval, it tended to be the same set of sensors whose reports would be overlooked, regardless of how significant the data they contained. The process of establishing the set of value beliefs which were to inform adaptation involved discussions between project system designers and track domain experts. The first stage was to establish a set of desirable high-level traits, e.g. processing should show preference to tracks in danger of being dropped, which were then subject to discussions addressing conflicts and trade-offs between behaviours. Three attributes were identified by which to appraise track state, track quality with an integer value range from 0 to 7 incremented or decremented after each scan cycle, track accuracy capturing issues of uncertainty and derived from a Kalman filtering process, and importance which is an operator assigned binary value. To reduce complexity, values were then classified giving three quality classes, two importance and two accuracy classes. The attributes were then subject to a series of pairwise comparisons to establish trade-offs in relative significance and the results combined to produce an overall ordering with values which were subsequently 'tweaked' by examining several scenarios. The final result of this value establishment process was a set of twelve classes each reflecting a different level of performance and represented by an initial starting value indicating their relative desirability, as perceived by the project members involved.

The time-value function shape, which identifies the temporal variation to the initial values during execution, was arrived at by defining t_c to be equivalent to the length of one sensor sweep cycle, i.e. 10 ms, after which the track was deemed to have zero value owing to its being superseded by newer arrivals. It was felt that if a track was processed in the first half of the sensor period, i.e. in the first 5 ms, then it would result in better performance than if it were processed in the second half, thus the slope of the graph was descending, see Figure 2.2. According to the authors, an implementation constraint arising from the choice of operating system meant that the graph had to be linearly decreasing and the final slope of the twelve graphs was arrived at empirically. The actual time utility associated with t_c for each function was not given but was illustrated in the outlined graph as being above zero, thus it is not possible to identify how tasks and their values interact as execution progresses (each thread handled one track and was assigned the value associated with that track). The goal of the scheduling policy was to maximise total accrued utility and the authors reported improvements under overload in terms of the average track quality and a reduced number of tracks being dropped, with no tracks classified as being of importance being dropped during their trials (a task would be dropped if it were not updated for a number of cycles). This does not necessarily mean, however, that all important threads/tracks would get to run in each cycle rather that the value function serves to 'catch' them when their status has deteriorated, and value increased, hence preventing them falling out of the system. No discussion of the overheads involved in the management of the scheme was provided.

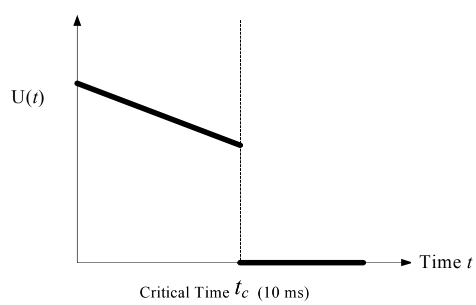


Figure 2.2: Established Time-Value Function for AWACS Tracker Example

This is an example where multiple attributes are used to assess the worth of an object with the treatment that object receives being dependent upon the measure of its worth. It should be noted that, having established the value at the arrival of the task there is no further need, in a time-value approach, to access them again as their behaviour is a known direct consequence of the temporal progression. This is in contrast with cases where the changeability of attributes is independent of time. Beyond the direct comparison of such metrics as missed deadlines, utilisation etc., an observer is confronted with certain difficulties when it comes to considering the effectiveness, or otherwise, of schemes such as these at a higher level. In the absence of a sound, theoretically proven, model by which to handle all aspects of this problem then taking an empirical approach is entirely justified. Likewise, if the project participants are satisfied with the value construction then, from a practical perspective minor criticisms of procedures are of little consequence. However, as the authors point out, the activities involved in identifying performance concerns and relative values, *"required considerable time and thought"*, and it therefore is beholden on them to justify the effort involved and to consider whether similar behaviours could not have been achieved by alternative and less time consuming processes. Some of the difficulties associated with definition arose because the participants found it a novel experience to think in terms of behaviour options in dynamic conditions and if a set of established procedures were to become familiar then this problem would be reduced with time. It is difficult for non-experts to query whether the choice of the original high-level traits was either the best or a valid one, and also whether the attributes chosen were the best or valid means by which to assess the issues. For example, would considering track importance alone have produced similar results at a far lower cost? Two immediately obvious facts are that, firstly, the values are devoid of meaning to all but those who defined them and the context within which they are defined, and, secondly, that subtleties such as a different mix of experts, with different experiences, or an extended investigation procedure, giving time for opinions to be modified, might all have resulted in a different set of values emerging and, consequently, a different set of behaviours being favoured. From the systems engineering perspective, it might be advantageous, for retrospective examination, to maintain a trail of thought processes, reasoning and preferences expressed during the creation of such mechanisms.

Since the original work of the mid-1980s, including that of Locke [Loc86], TVFs have made frequent appearances in the scheduling literature. However, the emphasis has been on the efficiency of the scheduling mechanism rather than on the semantics of what the value is supposed to represent. Indeed, whilst their ability to capture what Chen and Muhlethaler describe as the multivalent characteristics of tasks (that is soft and hard deadlines, optimal execution times and the like [Che96]) is frequently noted, little mention is made of how the actual worth of such functions is established and, more particularly, how such worth might change over time. The latter authors address the NP-hard problem of maximising the total value of tasks' contribution, by decomposing the task set into subsets which can then be ordered, introducing a scheduling heuristic which has a complexity of $O(n^3)$. Wang and Ravindran tackle the same problem and describe an alternative heuristic with complexity $O(n^2)$ which schedules packets on the basis of their TVF on an Ethernet-based networked system, used for supervisory real-time control [Wan04]. Packets are ordered on the basis of the ratio of their maximum possible utility to their deadline, which the authors refer to as the "pseudo" slope. This has the computational advantage of a much lower overhead when compared with the action of calculating the true slope dynamically. The derivation of TUFs (the term utility is used in this example) was assumed and no explanation as to what utility actually defined given. The authors found that, for the six TUFs studied (step, soft-step, linear, exponential, quadratic and composite) the Utility Accrual Packet Scheduling Algorithm they described was least effective for the stepped variants, i.e. those most resembling hard-deadlines. They also found that its advantages were reduced when handling traffic comprising smaller messages owing to the larger overhead requirements.

The advantage of employing TVFs to manage overload scenarios was again demonstrated by Richardson et al who applied time-value functions to communications in embedded real-time systems [Ric99] and a wireless network [Ric01]. The first example addresses the question of guaranteeing the behaviour for the sub-set of critical tasks by using fault prediction techniques to detect a fault and trigger the switch to the adaptive scheduler. In the second example the Wireless Adaptive Scheduling algorithm (WASA) employs a value function which considers the user's stated QoS requirements, specified in two parameters covering maximum probability of packet loss and maximum packet delay, together with a weight reflecting the reward or penalty associated with each packet's transmission and the degree of tardiness of a particular packet. When fluctuations in channel conditions give rise to a deterioration in performance then the per-packet value is used to adjust the packet's priority, effectively resulting in packets being ordered according to their importance and session state. ‘Importance’ is a reflection of the financial return to the system as a consequence of the successful delivery of a packet. Under normal conditions the Earliest Deadline First (EDF) [Liu73] scheduling algorithm is used to schedule packets and a switch is made to WASA only under times of stress. Richardson points out that when conditions are normal the operation of both algorithms is equivalent, that is they are both time-driven and synonymous. The authors raise awareness, in passing, of the issue of interaction between competing objects, in this case arising from the use of dual scheduling approaches, noting that even if only one session takes on WASA then it will interfere with all other sessions running under EDF. This is one of the potential difficulties which must be accounted for if flexible systems are to be developed which will make use of both value and priority driven scheduling for different sections of the workload.

Again, in [But95] the suggestion is made by Buttazzo et al that, owing to the performance differentials they have noted under different scheduling strategies, it might be advantageous to work towards developing systems which support multiple scheduling algorithms, each one geared towards a particular loading level. The authors found, in comparing several algorithms with and without guarantee mechanisms, that the most effective overall approach was to schedule tasks by deadline (EDF) and reject them on the basis of value. Under very high overloads, their Robust Highest Density First algorithm demonstrated a slight advantage, with tasks being scheduled on the basis of their value density, found by dividing their value by computational requirement. Value, in this context was defined as representing the relative importance of a task with respect to other tasks and is not discussed further. Therefore, it might be assumed to represent either a dynamically generated or statically assigned parameter. In

[But97] the author provides a further explanation of the nature and role of values as he sees them, stating that the value assignment is dependent upon the application and may be an arbitrary integer in a given range, equal to the task's computation time or the value density. However, when talking about values within the context of task level scheduling and real-time deadline constraints, he notes that value must also account for the task completion time too and that importance is, therefore, better described by a utility function (meaning a time utility function).

In general, not all values will represent properties which are time dependent and not all evaluation scenarios will focus exclusively on task scheduling problems. In using values to drop tasks, the advantage is that the pool of values from which the selection is made is known whereas in admitting tasks on the basis of value there is always the difficulty of knowing whether to accept the new arrival at a certain value and risk not being able to accommodate a future task with a higher value. Even in omission, as Davis notes [Dav95] there is a potentially detrimental side-effect in that it can lead to the dropping of several tasks which, individually have small values, but which collectively have a value which outweighs the newly arrived task with the highest individual value. Davis addresses this issue of accommodation with his Adaptive Value Density Threshold algorithm. In [Bar91] it was shown that for a uniprocessor system if the loading factor was ≥ 2 then no on-line scheduling algorithm can guarantee more than 0.25 of the competitive factor achieved by a clairvoyant scheduling algorithm.

In contrast to examples based upon time-value functions there are also those which demonstrate time invariant forms. Aldarmi makes the focus of his value a static interpretation of task importance [Ald99]. Another example occurs in [Biy88] which proposes the use of a *critical value* to control overloads in a distributed system where each node supports several processors. The critical value, taken from a 10-point scale, represents the task's relative importance and is effectively used to distinguish between essential and non-essential tasks. Procedures for establishing the relative order of criticalness and for allowing it to vary with context are not discussed. If a task arrives at a node and can be scheduled the value is not referenced. However, if there is insufficient capacity to accept a new task, the local scheduler checks to see if there are sufficient tasks of a lower criticality amongst those already accepted whose removal will make space for the newcomer. If this is the case, then a global scheduler re-allocates these elsewhere under a bidding-procedure [Ram84] and the new task is accepted, otherwise the new task is moved. In order to get round the previously mentioned difficulty of committing to actions which might subsequently prove costly, the approach is taken that the semantics associated with a scheduling guarantee are that,

> *"it is conditional upon the [subsequent] non-arrival of higher criticalness tasks which conflict with it"* [Biy88]

As with previous examples, the authors found that deadline based scheduling performed best when conditions were under-loaded whilst algorithms based on the critical value performed best under high loads.

2.2.2 Quality of Service and Soft Real-Time Processing

Part of the original motivation behind the introduction of value-driven operations into the real-time domain was a perceived need to account for the conflict between the deterministic temporal behaviour imposed on tasks, necessary to meet the desired predictability, and the actual role they fulfilled within a particular context. The use of hard deadlines negated tasks contributing towards system utility no matter how marginal their lateness. Also, there need be no correlation between task priority, which might be high, and their actual significance to the system, which might be low, meaning that when fixed-priority scheduling algorithms were employed, for example, high priority tasks would always be favourably treated regardless of the function they performed. With the increase in applications capable of tolerating a wide operational range and also in the growth of distributed systems, this focus on a single aspect of performance, namely time, and basing operations on pessimistic worst-case expectations of execution rates becomes unacceptable, leading to low levels of resource utilisation. Task and message management would need to become more flexible, or adaptive, and responsive in order that system resources might be allocated more appropriately and in a dynamic fashion. This led to the introduction of additional factors into the allocation process, e.g. network bandwidth and user-specified service tolerances. The parameters used to undertake such management activities can usually be collectively interpreted as "Quality of Service" issues, this being an umbrella term by which any variety of, generally non-functional, requirements can be expressed. Soft real-time processing and QoS-driven resource allocation schemes are frequently co-related in the literature. In such cases utility, with its ability to capture variable levels in behaviour, is also a frequently cited tool by which trade-offs between user/application requirements and system resources are undertaken [She95].

A leading example of theoretical work in this area is that of Lee's QoS-based Resource Allocation Model [Raj98][Lee99] which is centred on the use of utility functions to mathematically model the user's quality/resource trade-offs. In an attempt to handle the subject in an integrated fashion, it proposes solutions to a range of issues, from capturing the user's requirements through to describing two near-optimal algorithms with which to solve the NP-hard allocation problem. These collectively unite to create a QoS management framework which portions out resources with the objective of maximising the perceived utility to users. An ever-present design concern is that of bounding operational complexity. Lee classifies the problem along two axes: assigning either a single finite resource to multiple processes along multiple QoS dimensions or assigning several resources to multiple processes, again to satisfy multiple dimensions, and provides near-optimal approximation algorithms with which to tackle this problem. In order to capture the user's input Lee defines *quality indices* which map the quality space of any particular quality dimension onto an integer scale in a uniform way and which facilitate user specification. A dimension is a particular strand of quality concern, e.g. audio quality, frame rate etc.. These indices can then be used to define dimension-wise utilities with the sum of all dimension utilities forming the application's utility function. This translates to resource usage through the construction of a *resource utility* function. Thus utilities are employed in the general sense, not a time-variant form as previously discussed. Each dimension can be weighted for emphasis, mention being made by Lee that the weighting preference can be derived via the use of the Analytical Hierarchy Process, again, though this is not developed further. As Lee notes, the relationship between the quality of an outcome and the resources used to achieve it, is not functional, therefore dimensional utilities can be used to effect *QoS trade-offs*. System utility is defined as the weighted sum of application utilities, with weights purporting to represent some form of task importance, though this is not explored. Performance difficulties were encountered when the model was applied in large and dynamic systems, leading to a scalable, hierarchical variant being introduced in [Gho04].

Because of their generic nature, utility functions are a popular technique by which to quantify issues of concern, either multiple or single, and they appear frequently in connection with descriptions of adaptive systems. In [Par01] a utility-based model is proposed as the means by which to manage service level agreements in a network context, where utility is established as a consequence of discussions between user and service provider over the quality of service and acceptable costs of various connection levels. In [Kha97] and [Wan00] two more examples of utility being applied to adaptive multi-media systems are given, whilst in [Kel03] it is applied to resource allocation in a utility data centre. In the latter paper the definition of utility is deemed the responsibility of the user who is requesting certain permutations of resource allocation and it is the author's opinion that it is neither the concern of, nor within the scope of, the system designer's role to define these.

In [Cal00] the difficulty of resolving multi-attribute system design problems is tackled by employing a dual approach, first low level "value-oriented" descriptions of individual criteria using utilities, then, at a high level, following a "goal-oriented" methodology to compile the separate utilities presented by all the decision makers involved into an integrated representation of the overall belief structure. In noting the difficulties that utility-driven models have when presented with decision problems of a hard engineering nature, i.e. the very disparate characteristics and incomparable criteria, and also in trying to resolve discrepancies between several stakeholders/decision makers, they have indirectly demonstrated the consequences of either applying the wrong selection methodology to the problem or giving an inappropriate problem definition to the chosen method.

A scheme to enable adaptive applications within a wireless network, described in [Lia01], employs *bandwidth utility functions* to act as the mapping between the transmission bit rate and the level of service quality perceived by the application. The definition is provided by the user and, in conjunction with an *adaptation script*, which modifies the application's response in accordance with available bandwidth, they constitute the adaptation policy used. Again, utilities are quantized, so as to reduce operational complexity, and produce five discrete *critical utility levels*. The authors make clear that the actual value of a utility, of itself, has no meaning, and that it is only their relative positions which are of significance. Neither do they stipulate how the utility maps to the perceived quality in the application, though one suggestion might be by way of using industry standard techniques such as the Mean Opinion Score [ITU98], as they demonstrate in a previous work [Boc99b]. Despite the flexibility offered by using utilities, the difficulty of undertaking their evaluation in a real-time environment is raised in [Boc99a] where they were found to be problematic when applied to network level operations. In such situations compromises between the ideal evaluation of problems and that which can be practically accommodated must be made.

In [Kra99] the authors describe an adaptive media delivery service in which user-level QoS specifications, as represented by utility functions, are mapped onto packet scaling techniques to be used in a layered media scheme, where packets are dropped according to the priority assigned as a consequence of the expressed preferences (packets contain different types of media data). The two performance traits considered are those of temporal and spatial resolution, represented by attributes of frames per second and signal-to-noise ratio respectively. Reductions in the attributes' values represent reductions in the associated utility. As the authors point out, when considering the practicalities of translating user preferences to resource allocations or visa versa, such utilities are declarative only, that is they do not specify how this translation should be done. The user's utilities encapsulate the adaptation policy to be followed by the underlying mechanism and, consequently, by providing alternative graphs, whether actually as a consequence of user interrogation or arbitrarily arrived at, different policies can readily be effected. However, it is not clear at what point, or how, the authors envisage such user input being made in a real situation.

The continuous slope of a utility function, however, cannot necessarily be correlated to actual real-world states within the system, either because the system does not exhibit that level of granularity in the given operation or because it is too computationally expensive to work at that level. Therefore, having derived such functions it may be quantized, as seen elsewhere, into more manageable sections, as is done in [Cur03] where they reduce the number of performance levels to be accounted for. In this work, where the context is that of a time-aware admission control and resource allocation scheme for future generation mobile networks, the objective is to maximise system utility using user specified resource utility functions. Resource utility reflects the user's interpretation of what the resource, in this case bandwidth, will deliver in terms of application utility and that, in turn, can be summarised by the quality of the results produced. Allocation employs the convex hull optimisation algorithm, "asrmd1", as described in [Lee99]. However, the static nature of the utility function, which describes only bandwidth dependence, means that alone it cannot account for the dynamic progression of a process or for its temporal criticality. Therefore, the utilities associated with each connection, as provided by the user, were modified at each re-allocation point to account for such factors as the age of the connection and type of application. This demonstrates the more ad hoc approach to dealing with multi-attribute problems.

2.3 Discussion

Where knowing the state of criteria, and interpreting this state in a numerical form, is required then the evidence from the literature would point to the expression of value or utility functions being a favoured means of achieving this. However, this popularity would appear to be in spite of the stated difficulties researchers experience in applying them to theoretical, let alone, real world examples. The following comments are summarized representations of observations found in [Cal00] and [Chu98]:

- Where are utilities to come from as their derivation is a non-trivial process and users may be either unwilling, or incapable, of describing them [Wal04]? The problem definition and type of criteria to be addressed will have a part to play here as some issues may stimulate very strong and readily identified preferences whilst others are less easily exposed. In [Cal00] the author notes, for example, that criteria of a financial nature are more easily differentiated, i.e. trade-off points established, than many others might be.

Similarly, if probabilities are involved, as is normally the case when dealing with scenarios of uncertainty, then who is to define these, and how?

- The preferences of one decision maker will change with time and circumstances. The preferences of several decision makers may fluctuate even more. For example in [Raf02], where the problem scenario is one of software design, the authors found that the managers', i.e. decision makers', preferences changed as the project progressed. How is such changeability to be accounted for and at what point must preferences become fixed?

- The defined objective is repeatedly stated to be to maximise utility/value, that is either from the system or the user's perspective. However, as [Chu98] points out, if the evaluation is to be repeated and the integrity of the numeric representation is debatable, then it may be justifiable to select an alternative with a guaranteed utility of, say, 100, over one which returns a utility of 101 but has a reasonable probability of delivering a utility of zero. Also, as has been stated elsewhere, there is the potential difficulty of reconciling which perspective should prevail, the system's or user's, as the two need not, necessarily, coincide.

- The granularity of the measurement by which the criteria are assessed will influence the level of detail available for discussion but this may not lead to the process of capturing preferences or to the relative evaluations being any more meaningful though it will increase the number of relative treatments which the decision maker must consider.

Not only are these issues of relevance in the specific context of utility-based operations but they also have bearing on the wider problem of trying to build mechanisms which must integrate user preferences and resolve what are, essentially, multi-attribute problems.

With the focus still on value/utility, in addition to the previous observations several further issues might be raised. Although the term ‘utility’ may be the same regardless of context, it can be interpreted in different ways with work citing it as being drawn from the decision theoretic models, economic usage or, apparently, with no reference to any formal background at all. If the term is not clarified, then this may lead to different understandings as to what the purpose of utility is in any particular situation and, also, what the limitations upon its effectiveness might be. For example, Lee notes when comparing her work in the field of Quality of Service where she uses the terminology of ‘utility’ to that of Locke and Jensen, who talk about ‘time values’, that,

> *"Their [Locke et al] value function model is a utility function along the latency quality dimension of real-time tasks."* [Lee99]

leading an observer to question whether they are directly comparable descriptions and what are the different properties associated with each approach. In situations where utilities are left for users to define, would all users provide solutions which reflected the same interpretation? In addition, there is little by way of evidence of any analysis of whether the frequency of evaluation and the granularity of the level of operation is a suitable and supportable choice. In the cases where some input of preferential expression is required, the difficulty of extracting these in a consistent fashion has been mentioned. Another factor in this problem is that of the assumptions which those who are supplying the preferences are making about the context in which they are to be used; these may be incorrect or only partially applicable.

The examples provided can be broadly classified as dealing with either dynamic adaptive resource management or system/software design problems. When viewed in their entirety, these are inherently of a multi-criteria nature, but this does not necessarily have to be reflected directly in the method by which the problems are resolved dynamically as it may have been decided to deal with only a dominant criterion or decompose the problem into smaller parts whose resolution is dealt with at different points in time and in different ways. While utilities are very flexible tools, can the system they are employed in ever be flexible enough to cope with all potential scenarios? For example, the utility of a video conferencing service will vary with the nature of the business which is being discussed between participants, the duration of the connection, the frequency at which connections are made and so on. In considering the situations below it can be seen that the level of 'utility', taken in a general sense, will vary with the user, and there is likely to be more than one user involved:

- The first minute of a 30 minute image based presentation.
- The last minute of a 10 minute, one off, discussion with no visual content.
- The last minute of a 10 minute, daily, discussion with no visual content.
- The last minute of a 30 minute discussion, with high visual content.

A significant question for future systems is to identify the appropriate segmentation of such problems. This relates to the question of deciding at what point to "fix" the preference expressions, mentioned earlier.

Whilst the intent is to enable adaptation within the system, at whatever level, with the exception of the deliberation scheduling approach, no consideration is shown about the trade-off to be made between the form of the evaluation itself and the quality of results. The anytime model does not say how to effect an adaptable evaluation, rather they provide the framework in which it can be described. Not all decision problems are amenable to an 'anytime' treatment or deal with situations of uncertainty. Equally, having their roots in the domain of artificial intelligence implies an infinite succession of decision scenarios, where the precise definition of the problem is not known until it is encountered which, again, is not a universally applicable assumption.

2.4 Summary

The characteristics of multi-attribute decision problems, and the most appropriate way to resolve them if encountered on-line, will vary both with the type of problem and the context in which it arises. The risk associated with ignoring the particular characteristics which apply in any given situation is that they may lead to the adoption of unnecessary and inefficient forms of resolution. This chapter has presented examples of work which are relevant to this discussion and which illustrate the common prevailing treatment of what are, effectively, multi-attribute decision problems within a system context. It has illustrated how values are derived and how they are employed, the intention being to show that, although there is great potential for such usage, there is little by way of reasoning either about the inherent nature of the problems arising or about the appropriateness of the methods by which they are resolved.

Chapter 3

The Elements of Evaluation

Mention of such terms as ‘customizable’ or ‘quality of service’ in a system context, brings with it an implication of a choice in operations existing at some level otherwise realising such notions would be an impossible exercise. In turn, choice implies making a decision which requires access to a sufficiency of information if it is to have a beneficial outcome. Of course, a decision can be made upon the basis of a single criterion alone, such as cost or availability, but the complexity of systems and expectations of users, particularly in the area of consumer electronics, have increased to the point that in many instances this would no longer be an adequate response. A long-standing criticism of fixed-priority assignment algorithms, such as the Rate Monotonic algorithm, is that they only consider parameters based on either periodic rate or deadline and are, consequently, unable to take account of such factors as task ‘importance’. There have been several attempts at resolving this problem by devising new schemes by which additional criteria can be included into the scheduling operations but these, too, might be criticised for failing to provide sufficient reasoning and explanation as to how ‘importance’, and other such abstractions, can be derived meaningfully.

The problem which is the focus of attention here is that of how to approach the definition and implementation of dynamic multi-attribute decision problems and to do so in such a way that not only is the definition process made easier but also any flexibility in the definition can, at the same time, be exploited to enable the response to be adaptive. As with any process, the mechanisms employed must be adequate for the purpose intended. The motivations for exploring this area are several. There is a perceived need to allow for a greater degree of on-line resolution and customisation of actions by the system and its component processes, partly arising from greater software complexity and partly from the prevalence of networked systems.

Additionally, when discussing evaluation problems at a generic level, one must consider that such evaluations will be applicable in a variety of contexts and, therefore, should endeavour to keep the discussion as open as possible. In view of this, the relevance to on-line decision making of off-line traditional multi-criteria techniques may need considering further in order that their advantages, and disadvantages, can be highlighted.

This section outlines the rationale in pursuing this argument and introduces the core concepts involved together with relevant extant models. Ideally, there would be some correspondence between fast, frequent evaluations at a low level of abstraction using a small number of attributes and evaluations at a high level, running slowly, infrequently and needing to consider large numbers of attributes in order to resolve a critical decision. This may not, however, be the case.

3.1 The Decision Problem

The specific intent in employing dynamic evaluation is to enable the resolution of decision problems, ultimately leading to an act of selection. Given a set of alternatives and criteria by which to appraise them, Vincke identifies three forms which a multi-criteria decision problem might assume:

- Determining a subset of the alternatives considered to be best with respect to the criteria - a *choice* problem.
- Dividing the alternatives into subsets according to some norms - a *sorting* problem.
- Rank the alternatives from best to worst - a *ranking* problem [VIN92].

In terms of acts of choice, the above can be interpreted as: selecting a reduced number of the available alternatives which are deemed ‘better’ than the remainder; selecting partitioned groups of the alternatives; and selecting the alternatives on the basis of their position in an ordering of all alternatives. For the majority of decision problems an essential pre-condition is that an alternative exists, i.e. there are at least two items available for consideration. However, for the case where a single object is presented whether it merits an evaluation process being defined or not will be dependent upon the context of the problem, i.e. is selection to be made on the basis of a comparison against a global baseline, which raises the possibility of it being rejected. If there is only one candidate, and it must be used regardless of its state, then evaluation is redundant. The post-condition of evaluation is that each alternative has been assessed with respect to the decision objective. A choice problem can be represented as an action of *reduction* which would generally warrant the presence of at least one more alternative than is required if evaluation is to be justified. For a ranking action, the number of alternatives remains unchanged only post-evaluation they will be capable of exhibiting some form of ordering. In the latter case, therefore, the condition represented by the criteria should be the dominant motivator in subsequent usage otherwise, again, the ranking risks being redundant. For example, fixed-priority scheduling algorithms demonstrate selection at work in its purest form using a single criterion of task priority which reflects the dominant constraint in this context, that of time. Ranking on the basis of some other criterion and then still running on the basis of priority alone would bring little advantage. This is, of course, one of the dilemmas associated with use of such scheduling policies and it should not be forgotten that many other criteria which might be considered to be of relevance to the problem, have been informally handled off-line during the design stage.

It should be acknowledged that there are some differences to be expected between the kinds of decision problems normally illustrated in the economic and decision theoretic literature and those likely to be encountered in a dynamic, computer based, system. The former are single instance, generally complex and range in size between the very large scale and politically important, e.g. siting reservoirs, to the small and personally significant, e.g. buying a washing machine. They frequently feature a financial attribute and are associated with a variety of business scenarios, where the conflict between costs and quality may give rise to some disagreement between decision makers but makes for an easy interpretation of the problem. In contrast, it is likely that many of the multi-attribute decisions arising in a system context will exhibit some, or all, of the following characteristics:

- Highly defined with respect to the objectives of the process.
- Involving a constrained number of alternatives and attributes.
- Subject to operational constraints, e.g. deadlines and bounded resource usage, which perforce limit operations.
- Exhibiting a tendency for attributes to be of a dependent rather than independent nature. Additionally, the range of values which each attribute supports may be discontinuous.

These features make for a highly applied and concise selection problem when compared to those of other domains. There is also the anticipated difficulty of extracting the decision maker's value structure, particularly where the range of attributes being considered may reflect kernel variables and network constructs. Although there is no reason to suppose that decision makers cannot hold as strong a set of opinions about such things as with any other business-specific subject matter, it would seem reasonable to make it a requirement that those in a decision making role have a considerable degree of familiarity with the topic in hand.

A significant obstacle to carrying out effective decision making in a time constrained context lies with the practical difficulty of accommodating the necessary functionality within the allocated resources. In addition to this, there is also the problem of ensuring that the evaluation proceeds in a correct manner so that it reflects the belief structure which motivates it. A generalised decision problem is not normally subject to the restrictions found, for example, in time constrained systems. The time-scales employed are grossly different between the two and in the former do not necessarily constrain operations. Consequently, some of the mechanisms employed in decision theoretic processes may not readily translate to more controlled environments, despite their core importance to a discussion on the problem. There are several areas where difficulties may occur. The inclusion of criteria which have a subjective component may give rise to problems of how to capture and handle this aspect. The sheer number of alternatives and attributes specified may be difficult to process in the available time budget. If a functional approach is adopted, then defining utility and value functions which accurately reflect the stated preferences and reproduce the ordering these preferences imply is repeatedly cited as an activity not without its own problems [Gil05]. Likewise, with the specification of appropriate weighting constants [Poy97]. If theoretical decision models are employed to structure a resolution of the perceived decision problem then, in addition to implementation (space and time) constraints, designers must also ensure that the definition and final implemented solution fulfil the properties required by the model. This is of importance in two respects: firstly, it ensures that the values assigned to each alternative are an accurate representation of the factors they are supposed to measure; secondly, it ensures that a waste of effort is avoided. For example, the two axioms of significance in relation to utility theory [Kee76], an established method for the resolution of decision problems where the outcome is uncertain, are:

- *Preferential Independence* - that is, the relationship between attributes a_1 and a_2 should not be affected by the values held by remaining attributes.

- *Utility Independence* - that is, the utility function associated with an attribute is not affected by the values held by the remaining attributes.

These requirements may be difficult to guarantee unless they have been investigated in a rigorous and sound manner in the first instance. The consequence of not adopting a methodical approach is not that no usable result will be generated, but rather that values will be produced which will not be an accurate reflection of the underlying preferential model, which, in the absence of a process failing, will be nigh on impossible to detect. Such issues will need careful handling if they are not to undermine the integrity of dynamic decision making in a time constrained context.

3.2 The Evaluation Model

A multi-attribute decision problem is one where the ability to identify a solution to the defined problem requires the appraisal of two or more criteria, c_j, each of which explores a particular aspect of relevance to the problem. Each alternative object, a_i, belonging to the set of all available alternatives, A, is associated with its own instances of the criteria which are appraised in order to ascertain their individual conditions. Using this group of values the overall condition of an alternative can be interpreted such that the state of a_i, which represents its ability to satisfy the problem, is deemed equivalent to the combined state of its individual attributes, $\{ c_1^a, c_2^a, \ldots, c_n^a \}$. Thus, by evaluating each criterion, in a manner suited to its form, the individual results can be collectively summarised to represent the level of satisfaction associated with the choice of a particular alternative and also enable comparisons to be made with other alternatives evaluated under the same problem definition, such that if the *state of* a_1 "is better than" the *state of* a_2, then a_1 is more able to satisfy the problem. The fundamental difficulty with resolving multi-attribute decision problems, though, is how to undertake this process of collation in a meaningful fashion whilst also accounting for any trade-offs between the attributes. Disparity in units of measurement and their type rules out direct comparison. It is normally the case that gains on some criteria come at the expenses of losses on others, making the direct and unconsidered use of raw attribute measures potentially inadequate when it comes to considering matters of relative desirability within the set of alternatives. Such inadequacies can be remedied by the inclusion of additional information representing relative preferences for each criterion. This information must be supplied by the decision maker. Then, with both raw measurements and notions of preference, it should be possible to make an ordering of all consequences, or outcomes, and thus resolve the problem. There are several ways in which these two elements can be integrated, ranging from the formal definitions employed in the descriptions of value and utility functions, through less structured approaches based on attribute dominance and Pareto optimality, to ad hoc, and possibly arbitrary, treatments which reveal themselves in coding structures.

In order to obtain the necessary data certain roles must be fulfilled within the context of the decision problem. Figure 3.1 represents a high-level abstraction of the five main components involved:

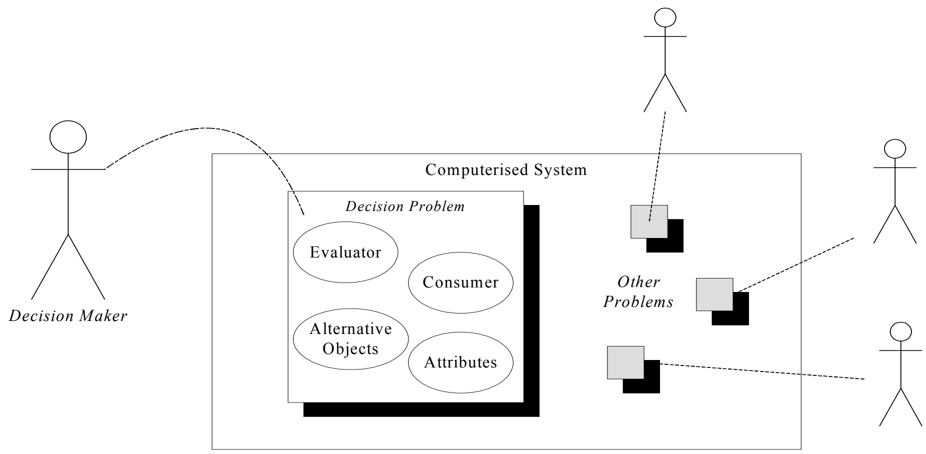


Figure 3.1: The Elements of a Decision Scenario

- *Decision Maker*: One or more human users who are interrogated to provide the belief structure which enables the relative worth of attributes to be established.
- *Evaluator*: The process which oversees the evaluation on-line, collates the attributes' values, runs the evaluation method and produces the necessary results of evaluation.
- *Consumer*: The process which consumes the results of evaluation. The consumer may have certain known requirements which are built into the problem definition but it should not have a direct input into how the attributes are addressed unless it is fulfilling the role of decision maker as well.
- *Alternative Objects*: The alternative objects available for consideration under this problem instance. Their number may be fixed or vary according to the state of the system and, at some level of abstraction, they will share a common characteristic, otherwise they would not be being considered under the same problem definition.
- *Attributes*: The criteria which comprise the subject matter of the problem.

It should also be acknowledged that, in some situations it may be appropriate to elicit user preferences directly during the evaluation of a problem. However, unless the user has full control over all elements of the definition phase they are not to be interpreted as the decision maker rather they are simply providing attribute updates. The key elements to be affected with respect to this issue are value structure, attribute and alternative definitions. Even if the user is allowed to select and reset all of these, if he is only allowed to choose from a finite number of options in the case of alternatives and attributes, then it is still arguable whether they are taking the role of exclusive decision maker or just sharing it.

The degree of active participation in the process of evaluation of each of the above components will vary with context, but generally the alternatives and attributes will assume passive roles, whilst the evaluator will assume the main dynamic role. Other functions and services may also be required and either subsumed within the above objects, or established as standalone components, depending upon the scale of evaluation being undertaken.

- *Attribute Update Process*: To undertake the process of updating the individual attributes. There is no assumption that the alternatives are physically associated with the attributes, nor that the attributes and alternatives are all located local to the evaluator. Therefore, it may be convenient to devolve responsibility to one or more processes local to the attributes themselves if their management is sufficiently complex to warrant it.
- *Data Repository*: Data is required in order to enable the management of the dynamic evaluation process. In addition to attribute and alternative object identifiers, information relating to attribute constraints and condition variables might be required, not to mention possible alternative evaluation methods and the material necessary to inform any adaptive processes. This latter category would include instructions to be followed on the handling of individual attributes together with details relating to the self-policing of evaluation frequency.
- *Adaptation Manager*: To identify the prevailing constraints on a specific instance of evaluation, both external (system) and internal, and to set the appropriate level of evaluation accordingly.

It is important to maintain a clear separation between the evaluation model in the abstract and its actual realisation. For example, no assumptions are made as to whereabouts within a system an evaluator is sited. All activities related to performing the evaluation might be contained on the local node or they might be dispersed across several. They might also be entirely static or random in nature. Such matters have no bearing on the integrity of the evaluation so long as they have been defined correctly and they do not deviate from that definition. They do, however, have a considerable bearing upon the practicalities of executing an evaluation which is an operational matter. They also have a bearing upon the definition process at the outset as they may be foreseen as presenting such a sufficiently challenging obstacle that they cause the original intentions to be modified, for example as when considering the set of possible attributes to define.

Returning to the issue of attributes, there is little advantage in being prescriptive about what can and cannot constitute an 'attribute' as it is impossible to delimit what will, and will not, be of interest to performance in any particular scenario. Examples of attributes and general discussions on the subject can be found in [Lar04][Bar95][Rav98]. It is important, however, that they exhibit two particular properties if they are not to devalue the investigation they are supporting. The obvious one is that, as agreed by the decision makers, they provide an adequate level of understanding of the issue they purport to represent. Even more obviously, they must be measurable [Kee76]. To this latter point might also be added the requirement that, for the purposes intended here, they are also accessible, as privacy and security limitations may preclude the ideal attribute being immediately available. Vincke references the work done by Roy and Bouyssou on establishing the "consistency" of attributes with respect to the definition of their properties [Vin92][Roy87b]. Consistency, in the general sense, however, is vital in both the definition of attributes and also in the execution of evaluation as a whole. Given the objective of finding that alternative which best satisfies the defined problem, then it is important that all alternatives are subject to identical treatment in any one instance of an evaluation otherwise the results will not necessarily be an accurate reflection of their relative worth.

Also, for these purposes, though this difficulty may arise in other domain specific decision scenarios as well, there may be a difference in levels of 'expert' knowledge between the decision makers and those responsible for realising the evaluation process, with the size of this discrepancy being dependent upon the particular problem context. For example, a client's specification for responsive software which supports several variants might suggest an appropriate approach being for selection to be performed on the basis of "system load", "user context" and "tool history". Even if these are subsequently precisely defined by the client, there is no guarantee that either they have sufficiently fully understood the infrastructure of the implementation, that at the point of specification they know what the infrastructure is to be, and have considered which, of the many possible interpretations of the above three attributes, are the best to implement. On the other hand, however, the actual programmers with responsibility for the application may have very clear ideas about how to set about this problem, only it may not entirely coincide with that of the client. This situation can be considered to be a *translation* problem and, thus, adds to the difficulties of the definition stage.

Depending upon the form of evaluation adopted further requirements may arise, as has been mentioned with regards to attribute independence for example. In the event that these cannot be met, or indeed, that there are any problems with the provisional attribute definitions it may always be possible to restructure the problem such that the intended attributes are redefined and the potential problem eliminated [Kee76]. It may be useful, in this respect, to distinguish two classes of attributes, those which undertake the direct observation of a property via the use of an attribute which cannot be decomposed further, e.g. the cumulative number of packet arrivals, versus those which employ meta-observations, that is they use attributes which address statistical and composite interpretations of low level primary data sources.

3.3 Dynamic Evaluation

When undertaken in a static context, resolving a decision problem is a one-off event. The problem is defined, the alternatives evaluated and the result produced. The complexities associated with the production of a solution can be offset either against the cost and/or importance placed upon the production of satisfactory results or against the longevity of the chosen outcome. By electing to handle decision problems on-line a range of additional factors must be considered on top of ensuring the sound definition of the problem. The term 'dynamic evaluation' refers to the on-line, and possibly repeated, resolution of a selection problem in which the attributes' values are appraised dynamically because they themselves are dynamic. This differs from static evaluation where attribute values are known off-line, the evaluation process completed off-line and with only the results remaining to be employed in on-line activities. Figure 3.2 outlines the architecture of a generic adaptive evaluation process. It follows that if the evaluation is assuming a dynamic role, in order to account for variations in the decision environment, then it too might be able to adapt to changes in the environment, thus increasing its robustness in the face of fluctuations in the system load. However, exercising dynamism will introduce new challenges to those already present when defining decision problems, and may conflict with existing approaches to the resolution of such problems. As Simon notes, when talking of the classical theory of Subjective Expected Utility [Neu47],

> *"Classical SEU theory assumes a fixed, consistent utility function, which does not easily accommodate changes in taste. At the other extreme, [decision] theories postulating a limited attention span do not have ready ways of ensuring consistency of choice over time."* [Sim86]

To illustrate the difference between dynamic and static evaluation, one can consider the case of static and dynamic priority assignment algorithms. The parameter used to schedule a task, in the case of the Earliest Deadline First method, assumes the value of the task's next absolute deadline, whereas under the Rate Monotonic approach it is a scalar value derived from the ranking of all tasks on the basis of their periodicity, shortest period first. Both approaches allow scheduling in a methodical fashion but exhibit different behaviours in doing so. With static priority assignment the entire evaluation process, from attribute measurement through to ranking of alternatives, is completed off-line, thus reducing the uncertainty and work required of on-line actions. In the dynamic method, however, the production of the scheduling variable is

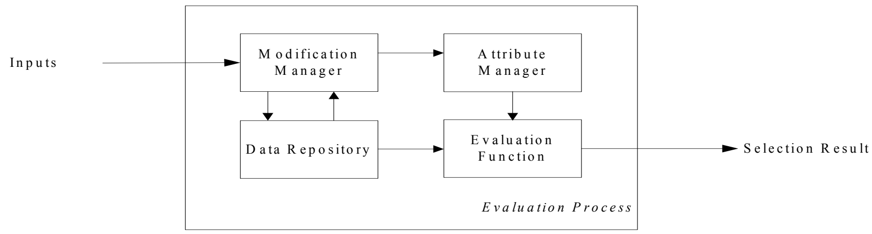


Figure 3.2: Generic Adaptable Evaluation Architecture

effected on-line using a completely different rationale. A critical difference between the two is that the rate monotonic value cannot be produced without the knowledge of all other tasks in the task set whilst the deadline value can be assigned independent of knowledge of the other tasks. This directly translates to their suitability for particular environments: for static systems the use of Rate Monotonic priority assignment avoids the need for repeated evaluation which would be unnecessary, whereas in dynamic scenarios, additions and removals from the task set are not a problem (leaving aside the question of upper bounds on viable performance) as evaluation is, itself, on-going and so can take account of these changes. This example also shows that different attribute definitions can still be used in support of the same activity.

By enabling evaluation to be dynamic it is also possible to expand the scenarios in which it can be employed. Static evaluation can only be employed in cases where the number of alternative objects is fixed and known and where the attributes' values are static and known. In all other cases where either, or both, of these characteristics are unknown, then unless on-line evaluation is supported the evaluation cannot be performed with the possible consequence that the functionality of the parent process is permanently constrained by limiting assumptions.

A potential pitfall which needs to be taken into account is that of distinguishing the causal factors when performance fails to meet expectations. Validating preferentially-driven operations can only ensure that the evaluation, as specified, is implemented correctly. It is not capable of exploring whether the fundamental assumptions which motivate the evaluation are correct. Indeed, it is a basic truth when handling decision problems that even though the use of proven tools can help you arrive at a solution to a problem in a consistent manner, they can never be used to argue that the identified solution is also the absolute best outcome possible. Decision tools do not find the theoretical best solution but only that which is optimal, in light of the stated criteria, from the set of alternatives under consideration. Thus, an evaluation may perform correctly but the outcome may still cause dissatisfaction if the decision maker's preferences were not accurately revealed.

A second problem can be safeguarded against by implementing appropriate exception procedures to deal with situations where attributes and objects are temporarily unobtainable leading to the possibility of inconsistent treatment. There is also the difficulty of distinguishing between the situation where behaviour is inaccurate, possibly because of the previous issue, and it is accurate but results fall outside the expected range. For example, if the assumption is made that there will always be a usable result returned, that is the given requirements will always be met at least to the minimum standard, then what will be the impact on functionality if, on one occasion, they are not, e.g. all alternatives have a value of zero? Also, how can an observer of the system, or indeed the actual consuming process, know whether a zero value represents the real state of the alternative or is due to an operational mishap?

3.3.1 Overheads Associated with Dynamic Evaluation

In discussing the possibility of employing dynamically appraised value-driven operations it is necessary to raise awareness of a significant obstacle in the path towards its acceptance, namely the additional burden on performance which this approach implies. If an evaluation is to assess non-static criteria then it must, itself, be non-static i.e. evaluate the criteria at a frequency identified as relevant. The question of relevance is settled by consideration of the particular circumstances of the evaluation problem and refers to the issue of whether the evaluation is driven by the evaluating process, i.e. when a point is reached in the process at which values are required they are revised/produced regardless of attribute state, or whether it is driven by the changeability of the attributes which characterise the problem. If the latter were the concern then, theoretically, evaluation might be performed continuously whilst the problem remains unresolved to account for changes in attributes, an action which would clearly undermine efficiency.

To illustrate the variations in behaviour which are a consequence of undertaking dynamic evaluation, a scheduling example is provided in which a group of ten tasks is scheduled using a dynamically derived value (task details are shown in Table 3.1). In order to reflect the behaviour of the Rate Monotonic and Earliest Deadline First scheduling algorithms which were used as controls, the evaluation was performed at the point of release, i.e. a value assigned to a new task, and subsequently at each point when a task had completed, i.e. existing values revised in order to find the next ‘best’ task to run. Under the EDF and RM methods pre-emption of running tasks is caused by new arrivals only, whereas with value-based scheduling, depending upon the relationship between the frequency of evaluation and task periods, a task might be pre-empted either by a new arrival or by a waiting task when the re-evaluation brings about a

Task ID	*Period*	*Computation*	*Rate Monotonic Priority*	*Number of Released Tasks*
1	10	2	1	100 000
2	15	2	2	66 667
3	20	4	3	50 000
4	25	2	4	40 000
5	38	3	5	26 316
6	45	4	6	22 223
7	60	8	7	16 667
8	75	3	8	13 334
9	80	2	9	12 500
10	100	2	10	10 000

Table 3.1: Task Characteristics

change in the tasks' relative status. In this trial, re-evaluation (of all active tasks rather than just newly released ones) was only performed after a task had completed so once selected to run, a task could be pre-empted solely by new arrivals but a new arrival need not necessarily have the highest value and, therefore, become immediately runnable. A counter was maintained on each operation of value change in order to illustrate, at an abstract level, the amount of work which would be required to maintain values in a real system. Likewise, the number of pre-emptions was also counted, the objective being to show the fluctuations in winning objects arising from the frequency of evaluation and from the changeability of attributes. In scheduling, pre-emption is associated with the specific cost of swapping tasks in and out and so in other problem scenarios such costs would assume a different form.

The evaluation method used for the production of the value was a weighted additive function. However, instead of being thoroughly derived from real preferential beliefs, the weighting constant, whose total value was 1, was equally divided across all attributes, e.g. for two attributes each had a weight of 0.5, for three 0.33 and so on. Each of the value-based variants reflected the use of a different number of attributes ranging from 1 to 5 in total. *Attribute 1* was assessed by a value function which interpreted the amount of slack left to the task, thus it represented a temporal component and received consistent treatment across all tasks. *Attributes 2* to *4* were randomly generated floating point values which changed each cycle independent of when evaluation was undertaken and so returned a new value each time they were referenced. They were intended to represent equally dynamic real-world criteria.

The purpose in providing this example was to illustrate how changes in value, arising from the reasoned employment of a dynamic evaluation process, have an impact upon operations. Value-based scheduling, as illustrated here, is an example of a multi-attribute, high frequency dynamic evaluation scenario in contrast to RM and EDF both of which are single attribute evaluations, the former being a one-off static evaluation whilst the latter can be seen as an open-ended dynamic evaluation.

The total utilisation of the task set was 0.9995 which gave rise to task failures under the Rate Monotonic approach. The scheduling simulation was run for 1,000,000 cycles and a periodic task model was assumed with tasks being aborted if they had not completed by their deadline. It must be stressed that the scheduling metric of a missed temporal deadline, by which performance is commonly assessed, should not be used to evaluate performance in this instance as the assumed evaluation/problem definition did not explicitly state this as the primary criterion. Rather, the problem definition might be summarised as "while tasks are within their deadline, schedule that task with the highest value". Therefore, the number of completed instances of each task, where completion is an act constrained by time, is used to indicate the differences between the scheduling variants rather than as a criticism of the efficacy of the particular scheduling scheme being employed.

Overall the total number of completed tasks for the five value-based variants decreased as the number of attributes increased, as shown in Table 3.2 (the case where the value is based on 1 attribute is something of an exception as *Attribute 1* represented the remaining slack and was akin to EDF in terms of its behaviour). This is to be expected as the dominance of the temporal attribute was increasingly undermined as a greater proportion of the overall value became dependent upon the other attributes. However, some individual tasks performed better under the value-based approach than they did when scheduled by the fixed-priority algorithm, as can be seen from Chart 3.1. Indeed, in general the top-half of the task set faired worse under the value-based approaches as they had smaller periods so the amount of slack available was lower and the leeway for making up occasions when interrupted reduced. Tasks 8 to 10 in the bottom half showed an improvement owing to their longer periods giving them increased opportunities for being promoted. Chart 3.2 shows the average number of value assignments experienced by each completed task and, as might be expected, the figures were much higher for the value-based variants. For those tasks with longer periods, the number of value assignments per completed task tended to decrease as the number of attributes employed in the generation of the value increased, an effect largely due to the decreasing influence of the time-based attribute (*Attribute 1*), therefore permitting these tasks to finish earlier than they might otherwise. Chart 3.3 displays the average number of pre-emptions experienced by completed tasks across the

	Rate Monotonic	*EDF*	*Value = Att 1*	*Value = Att 1 & 2*	*Value = Att 1, 2 & 3*	*Value = Att 1, 2, 3 & 4*	*Value = Att 1, 2, 3, 4, & 5*
Completed Tasks	349224	357707	357705	343270	336709	333301	331470
% of Releases	97.63	100.00	100.00	95.96	94.13	93.18	92.67

Table 3.2: Total Number of Completed Tasks under Different Scheduling Approaches

different scheduling schemes. The peak around Task 7 reflects the fact that it had the largest individual computation requirement of the task set. With the exception of Tasks 1, 2, and 4 which had the shortest periods (Task 3 had a higher ratio of computation to period length), the purely time-driven scheduling approaches gave rise to a greater number of pre-emptions per completed task than the value-based schemes, as summarised in Chart 3.4 which represents the total pre-emption and value assignment ratios for the task set. Again, this is due to the decreasing influence of time within the value. In addition to being influenced by the amount of computation carried out, pre-emption figures are also affected to a degree by the harmonic nature, if any, of the task periods which might result in new arrivals coinciding with tasks completing and thus making pre-emption unnecessary.

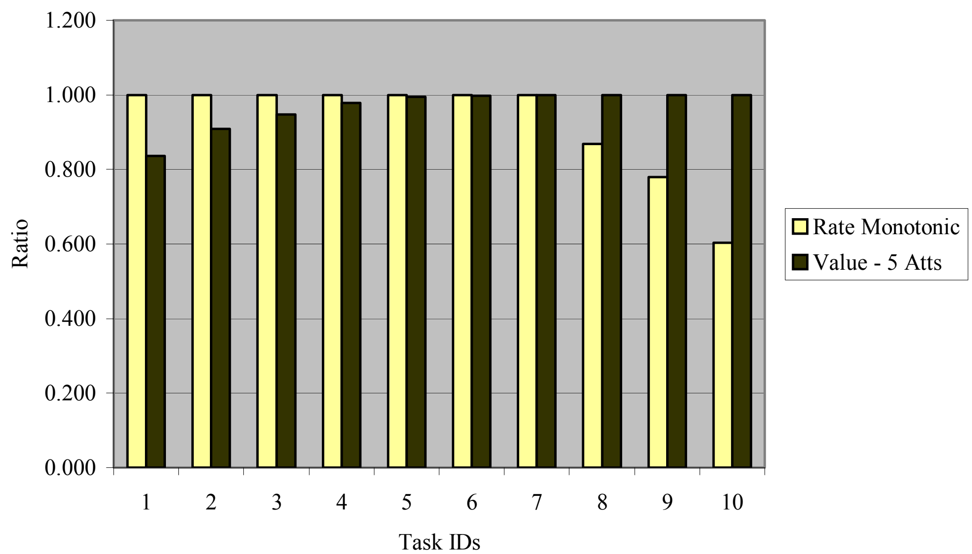


Chart 3.1: Ratio of Completed Tasks to Released Tasks under Rate Monotonic and Value-Based Scheduling Variants (using 5 attributes)

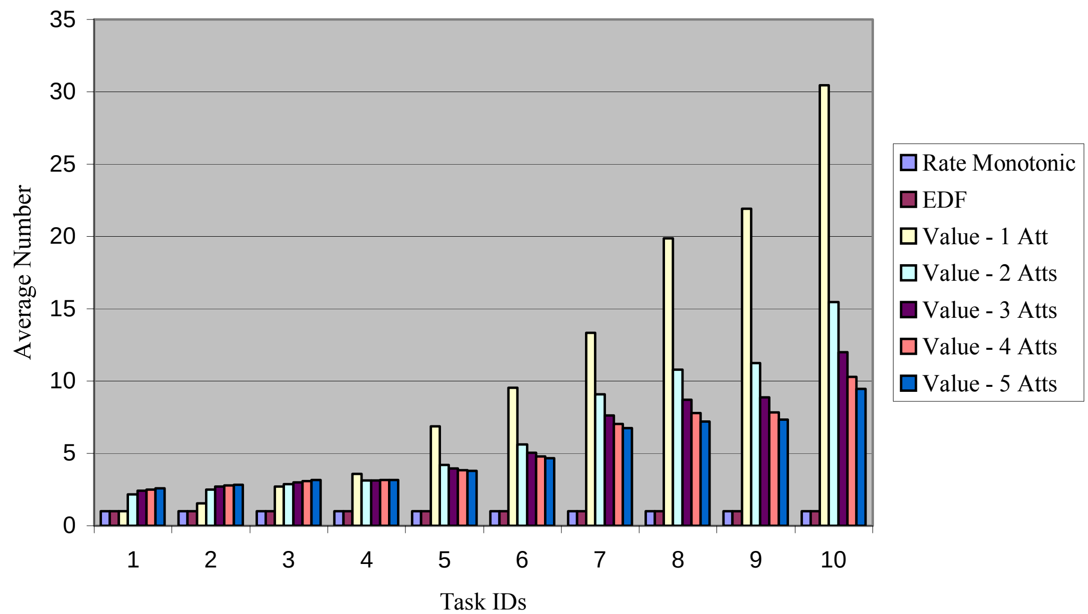


Chart 3.2: Average Number of Value Assignments per Completed Task

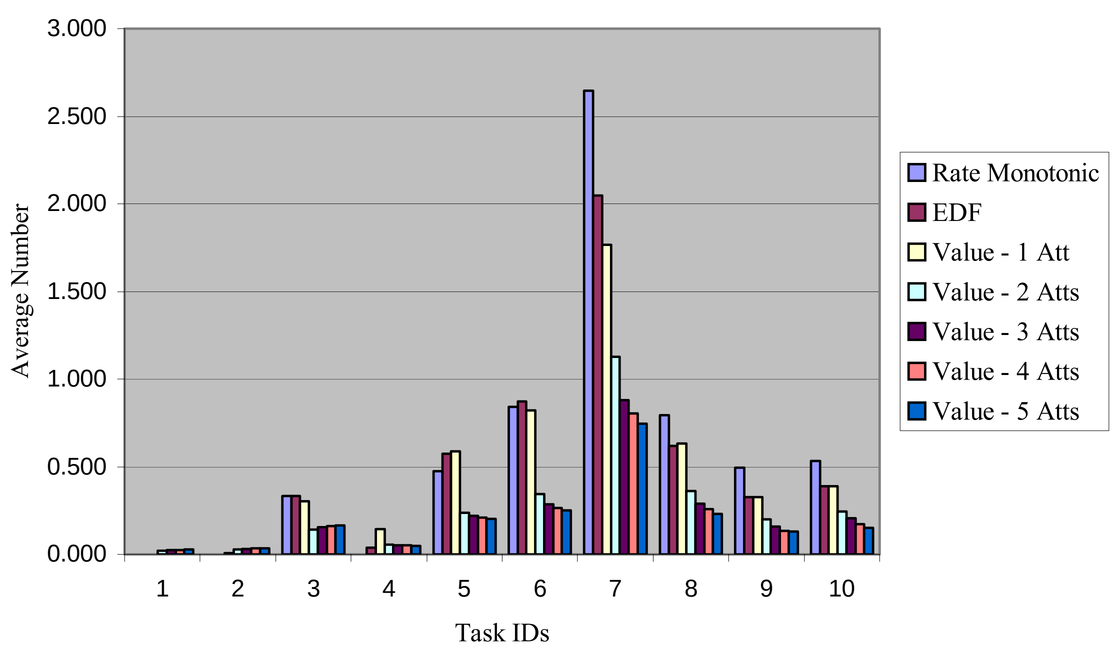


Chart 3.3: Average Number of Pre-Emptions per Completed Task

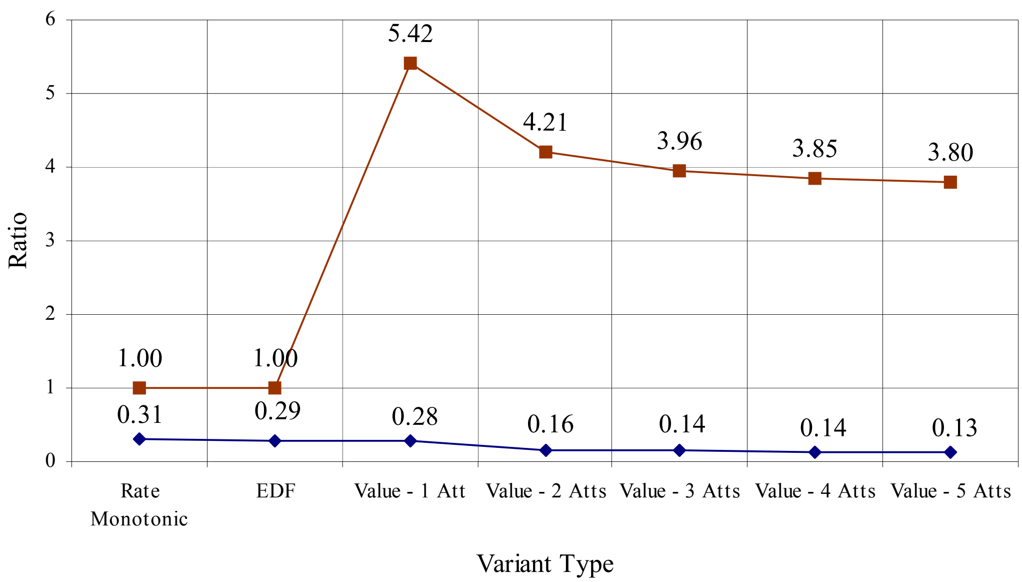


Chart 3.4: Ratios of Total Number of Pre-Emptions and Value Assignments to Completed Tasks

The inference which is to be drawn from the above example is that the definition of the evaluation, in terms of number of attributes and frequency of evaluation, must be considered carefully as the choices made impact directly upon the results produced. This being said, however, it should also be noted that the behaviour of the value-driven schemes was not wholly unpredictable, as might have been initially assumed if contemplating value-based operations. Indeed, the value-based scheduling scheme illustrated here can be interpreted as demonstrating the inverse set of failings as those seen under Rate Monotonic scheduling. In the latter case, tasks are penalized under high loads for having periods within which to complete their work that are 'too long' (that is their computation to period ratio is low) whereas under value-based scheduling, they lose out if their periods are 'too short' (or their computation to period ratio is high). Where the decision problem assumes the form of task scheduling, then one conclusion which follows from the above is that the effectiveness of the evaluation methodology, which results in the value (scheduling parameter) being produced, needs to be explored specifically within the context of the particular problem scenario it is intended for as such schemes might prove to be more than adequate for some uses and highly unsuitable in others. This is, of course, to be expected but it would be advantageous to establish which of the possible evaluation methodologies under consideration for any particular evaluation scenario actually provides the most satisfactory solution.

3.4 Meaning and Role of Value

3.4.1 Single Value Form

The advantage of combining the multiple concerns represented by attributes into a single value by which to describe each alternative is that it facilitates comparison between a set of objects. This feature is of particular relevance as the number of alternative objects increases. Not only are the relevant positions more clearly seen but the act of comparison can also be undertaken more quickly. From a practical view, there is also the consideration that a result in single value form reduces the size of data required, both for transmission, and storage purposes. There is the additional factor that it offers a certain psychological advantage to users reassured by the existence of a succinct numeric representation of a solution to a problem. However, insisting upon a single value format for results introduces the problem of finding a method by which this can be achieved in a valid manner and which accurately represents the underlying value structure. The question is how much effort is it worth expending in specifying a problem in order to achieve this end? Speaking of a similar multi-criteria scenario and attempts to encapsulate a single value representation of 'quality' in a software engineering context, Boehm commented that,

> *"Finally, we concluded that calculating and understanding the value of a single overall metric for software quality may be more trouble than it is worth. The major problem is that many of the individual characteristics of quality are in conflict; added efficiency is often purchased at the price of portability, accuracy, understandability, and maintainability; added accuracy often conflicts with portability via dependence on word size; conciseness can conflict with legibility. Users generally find it difficult to quantify their preferences in such conflict situations. Another problem is that the metrics are generally incomplete measures of their associated characteristics."* [Boe78]

Examples of work where the numeric interpretations of qualitative features are proposed can be seen in [Hol98] and [Kim01]. Employing a single value to represent more than one property of an alternative object has been an approach followed in the real-time community too when attempting to expand scheduling models so as to account for more than just timing information in task behaviours. Again, in this situation encapsulating several issues into a single value form enables the scheduling activity itself to be less complex in its operation.

The use of the term 'value' is somewhat misleading as, of course, every action in a software-based system is value led. In this context, however, it should be understood to represent a variable derived from the appraisal of two or more attributes on-line. As with any dynamic variable, a value will be associated with a representational range and, of itself, is inherently as predictable as its peers. Even its subsequent usage may present no threat to the certainty of operations if, for example, it is employed to select a fixed number of alternatives from a group which all have the same operational requirements. It is worth reiterating, however, the caveat mentioned in Chapter 2 regarding the uncertainty which can arise with respect to the use of terminology in this area. To repeat, it is common to see references in the literature to the use of value and utility functions but with little indication of whether they are consistent in their usage with other examples. If venturing down the route of employing the dynamic evaluation of decision problems attempts must be made to ensure any such misunderstandings are avoided.

3.4.2 Value Interaction

Another factor to consider when employing dynamic evaluation is that the specific form of the evaluation may not, on subsequent inspection, produce quite as much variation in results as had been anticipated. Such behaviour is a consequence of the particular nature of the values produced together with the use of attributes which display different degrees of dynamism and calls into question whether an alternative form of interpreting conditions might be better employed. Take for example the case where each attribute's measured value is modified by a weight, and where each attribute is assessed through the use of value functions which return scores within the range [0,1]. The alternative objects are appraised through the use of two attributes, *A1* and *A2*. *A1* might represent either some static property, or a relatively stable one which exhibits uniform increments during the object's life, and has a constant minimum value with respect to each object. *A2* represents some highly variable property whose minimum value may be 0.0. Given that the weights applied are approximately similar, in this case 0.5 each, then this means that the existence of *A1* ensures that each object has a guaranteed minimum overall value. It may be that, no matter how high an object scores on *A2,* it will never be sufficient to impact upon the order of values which would otherwise have been seen by using *A1* alone.

Figure 3.3 illustrates this pattern for the case where *A1* on *Object 1* has a guaranteed value of 0.3233 whilst on *Object 2* it has a guaranteed value of 0.4596. In this case, this means that *Object 1* has to achieve at least 0.1363 on its *A2* in order just to match *Object 2*'s minimum status. If *A2* on *Object 2* has a value which is greater than (0.5 - 0.1363), i.e. 0.3637, then it will win regardless. If this scenario is encountered every instance of evaluation then, irrespective of what state *A2* is on *Object 1*, the latter will never win. Of course, the result would be a correct interpretation of the decision maker's beliefs as specified by the evaluation method. However, if

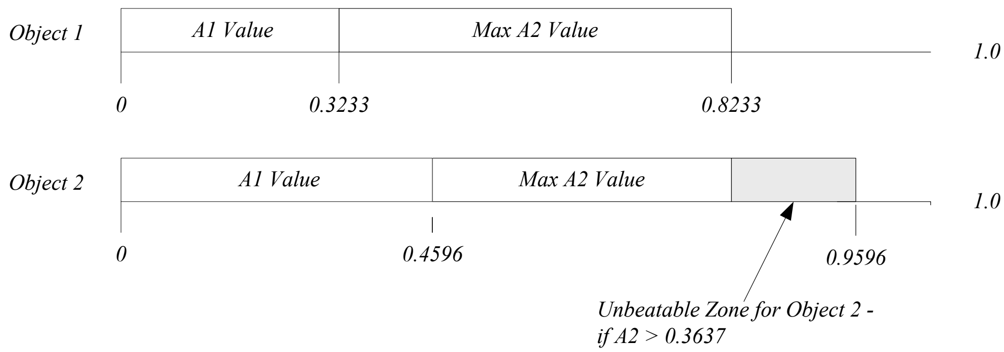


Figure 3.3: Illustration of Restrictive Behaviour Associated with Particular Value Combinations

the degree of dynamism in the attributes has been misconstrued then dynamic evaluation of this form may be largely a redundant activity as, on most occasions, the dynamism of some attributes cannot compensate for more consistent behaviours in others. Additionally, without giving thought to the structuring of value forms, users' faith in the ability to capture and respond to the dynamic nature of an evaluation problem may not be entirely justified.

Another consequence of the interaction of value ranges, namely variations in the reasons why a winner has emerged on a particular occasion, might also be mentioned because it serves to illustrate that notions of ‘predictability’, in the general sense, may need to be relaxed in the context of dynamic evaluation. The observation is based upon an evaluation using the weighted additive model with the two attributes of the previous example, one attribute representing a time variant property and one a random property, but it may also be applicable in other scenarios. An object is selected owing to its having the ‘best’ value amongst the available alternatives, thus selection of a ‘winning value’ is dependent upon the performance of its own attributes relative to those of all other alternative objects. However, the interaction of criteria, and their mutual relationships, facets which govern their contributions to the final value, vary with each evaluation instance. This means that, in the case where a single winning object is required, the significant contribution which secures the winning position may come from an object having any one of the following: the maximum value for *A1,* the maximum value for *A2,* high values for both attributes and having relatively mediocre values for both attributes which, combined, manage to outperform all other contributions. This is the effect of trading-off attributes and compromising on their perceived worth but these behaviours are hidden in the final value leaving the consumer of results uncertain as to the precise make-up of properties in the winning object's state.

3.5 The Meaning and Role of Preference

A reason for explicitly handling the abstraction of preference has already been given when stating that it is a necessary element in the satisfactory resolution of multi-criteria problems where trade-offs of some form are required. It is also beneficial in that it facilitates the evaluation of problems containing mixed types of attributes, particularly those of a non-quantitative nature and it is suggested that it is fundamental to the satisfactory resolution of any problems where the objective is specifically geared towards the resolution of issues of ‘quality’. Preference is a useful abstraction by which to represent issues of subjectivity within system operations and, whether explicitly handled or not, it will be exercised at many points within the design process. For example, the act of deriving weighting constants, no matter that they are identified informally, is an expression of preference which favours or penalizes the variables with which the weights are associated. Orderings on groups of objects reflect varying preferential treatment as does the selection of components and tools. In many such instances it might be considered beneficial to an understanding of system operations if such preferential behaviours were noted.

The significance in considering preference per se as a motivator for human and systemic behaviour is that it may be used to account for frequently observed actions which otherwise might lack an explanation. A properly defined set of preferences should, in principle, lead to a consistent and repeated interpretation of any situation. However, unlike the domain of economics where there is the proviso that the behaviours which are supposed, theoretically, to be exhibited, may not actually be observed in reality, in a system's context any deviations in behaviours from those prescribed by the model can be considered an error.

To take a high-level illustration of the impact which hidden, and non-expressed, preferences can have on the efficiency of operations, in this case those associated with human end-users: two printers are networked and are identical in every respect apart from their network address and physical location, one being at the bottom of a flight of stairs whilst the other is situated at the top. The system view is one of equivalent machines, however, the actual users may largely prefer the lower machine on account of its being physically easier to access. Over time this will lead to an imbalance between the two printers in terms of repairs, supplies and maintenance requirements which the system will be unable to account for as the subjective property of ‘perceived ease of access’ has no representation in its operations.

The action of 'trading-off' one attribute against another reflects the relative desirability of each type of attribute and also, possibly, the value it currently holds too. A set of preferential beliefs may originate from highly subjective opinions of the human component in the design process, political/economic factors, legacy issues in design and product development, business policy, functional and non-functional requirements. High-level preferences (user, business etc.) may be reasonably self-evident and, consequently, relatively easy to capture. However, applying the concept at a low level, where they may be buried within the informal activities of software engineers, may be more difficult.

If preferences are to be considered in more detail, then an initial step is to identify what the preferential beliefs are for a given context. Established techniques used to interrogate users in the fields of decision and economic analyses are potential sources of such information but must be applied correctly. It is suggested here that all expressions of preference arise ultimately from a measurable, or observable, difference between elements even though the decision maker may not, himself, be aware of the causal factor. If this was not the case then exercising preference would be equivalent to making a random choice. Unfortunately, this difference will generally be remote from the current context in terms of both time and space meaning that it may well be beyond the scope of the system to detect and infer the nature of any such preferences.

Within an evaluation, trade-offs between attributes is realised through the use of such techniques as weighting constants and utilities which allow for preferential treatment to be focussed upon a particular criterion. Additionally, preference can also be exercised indirectly through choices made at the design stage as to how, and what, components are represented.

3.5.1 Theoretical Model of Preference

An established preference model exists by which to accurately capture the above notions [Vin92][Ozt03] based upon binary relations between pairs of alternative objects. This is used in decision theory to help refine the decision problem and thus facilitate the finding of a solution.

Given a set of alternative objects, $O = \{ o_1, o_2, ..., o_n \}$, a binary relation R is expressed for all pairs within the set, such that R takes one of the following forms:

- is *preferred to,* that is $o_1 \; P \; o_n$
- is *indifferent to*, that is $o_1 \; I \; o_n$
- is *incomparable,* that is $o_1 \; J \; o_n$, either because they are not alike or there is insufficient information available to perform a comparison

In order for these relations to accurately represent notions of preference it is assumed they exhibit the following properties:

$\forall\ o_i \in O$

o_1 P o_n $\rightarrow o_n$!P o_1	P is asymmetric
o_1 I o_1	I is reflexive
o_1 I o_n $\rightarrow o_n$ I o_1	I is symmetric
o_1 !J o_1	J is irreflexive
o_1 J o_n $\rightarrow o_n$ J o_1	J is symmetric

If, for each pair of alternatives, one and only one of the above relations holds, then the relations define a *preference structure* upon *O*. Alternatively, a less strict *preference relation, S,* describes the first two properties and this can then be employed to help find a solution to a decision problem, as mentioned earlier [Vin92]. An objective in decision theory is to find a function, *f*, which optimizes one or more criteria defined on *O,* to produce a mapping from *O* to a numerical value set such that *S* is preserved and the magnitude of the values identified used to resolve the problem,

if $\quad o_1\ P\ o_n$ *then* $f(o_1) > f(o_n)$

$\quad o_1\ I\ o_n$ *then* $f(o_1) = f(o_n)$

A traditional preference structure produces a mapping which is defined as a *complete,* or *total, preorder,*

$$o_1\ S\ o_n \Leftrightarrow f(o_1) \geq f(o_n)$$

and it allows for the elements in *S* to be ranked from 'best' to 'worst' with the potential for ties. If *I* is limited to identical ordered pairs, ruling out the possibility of equivalencies between two different elements, then the preference structure describes a *complete order*.

Additional orderings describe different forms which the preference structure might assume and enable a greater level of realism to be entered into the model. A *semi-order structure* is one which models situations where indifference between alternatives is dependent upon a sensitivity threshold, above which indifference is replaced by a positive preference for one of the alternatives. An *interval order structure* reflects a similar situation, only in this case the threshold is variable rather than constant. The orderings described so far have assumed that there are no incomparable elements. However, this is seldom likely to occur in a real example, and similar *partial* ordering structures which focus on sub-sets of *O* can account for scenarios including incomparability if necessary. The alternative is to redefine preferences in such a way that they become a complete pre-order structure.

3.5.2 Value, Quality & Preference

The concept of 'quality' is not amenable to a single definition. One only has to consider the frequently used phrase 'Quality of Service' to realise the difficulties of providing a precise description of the term and one which is applicable to the myriad of contexts in which it is employed. As a general rule, however, any satisfactory resolution of a quality-driven problem must include an appraisal of the two elements:

- *Sufficiency* - representing a quantitative assessment of the physical components within the problem, showing that there is an adequate substance to the components in the solution.
- *Preference* - representing ostensibly intangible subjective issues relating to the relative desirability of components within the problem.

The role of the 'value', therefore, is to allow the combined representation of these two elements within operations. One of the advantages in considering these issues is that they provide a useful tool by which to justify some actions.

3.6 Summary

This chapter has introduced the core elements central to a discussion on dynamic evaluation and given some indication of how the roles of value, quality and preference are intertwined. It has also provided illustrations of possible mechanical difficulties particularly associated with the employment of a single value form of numeric representation.

The standard description of what constitutes a decision problem, as given in Section 3.1, illustrates the universality of their characteristics. However, it was suggested that the types of decision problems more likely to be encountered in a real-time context would differ slightly, for example by being more scoped in their subject matter and having a greater level of dependency between attributes. Such differences prompt the question of whether existing decision resolution techniques are appropriate ways to tackle dynamic evaluation problems. The purpose in suggesting the deployment of dynamic evaluation within the context of a real-time system is twofold. To introduce a time constraint into the performance of an evaluation, hence it cannot be considered an open-ended operation but instead must be completed and return a result within a given time frame. Also, because of its relevance to certain classes of user and system process in this area, e.g. soft real-time media or distributed data processing applications, where the ability to carry out dynamic choice and selection functions might be deemed an asset by which to enhance self-management.

Structures must be identified, however, to help realise dynamic evaluation mechanisms. The description given of a generic evaluation process followed the common attribute-based model with the favoured term for describing the objective being the appraisal of 'value'. Consideration of risks associated with the acquisition of alternative outcomes was explicitly excluded from discussions, ruling out the necessity for subsequent investigations to consider issues of the probability of events. In summary, the approach described in Chapter 3 focused on three particular aspects of evaluation design. Firstly, it made it a requirement that the chosen evaluation methodology produce results in a single value form, i.e. the composite state of each alternative with respect to the problem criteria is given a single numeric representation. It is assumed that the operational advantages to be gained from only having to handle one value per object will outweigh the stated difficulties in their production. Secondly, it advocates that a greater emphasis be placed upon the explicit handling of notions of preference within system operations. Not only are subjective issues of preference a key element in attempting to appraise quality in general and in distinguishing the relative merits of various 'quality-driven' operations, but they also play a fundamental role in being able to identify the form of a satisfactory solution to any decision problem. Thus, this abstraction is deserving of greater attention. Thirdly, it highlighted several roles which need to be fulfilled within the evaluation process. This is in contrast with static decision scenarios which do not decompose the problem in this fashion. Dynamism, however, gives rise to problems which do not occur in the static case, such as adaptation, and the system assumes roles that, in the real-world, are subconsciously filled by human entities who have the ability to modify and intercede into the decision process as it progresses.

The discussion relating to performing an evaluation was addressed at a generic level. The solution to a known decision problem is arrived at by the interrogation of two or more attributes which characterise issues of relevance to the problem. As a consequence of measuring the attributes, the relative state of the available alternative objects can be established. There is no advantage in endeavouring to specify particular evaluation mythologies, suitable attribute types and appropriate numbers of alternative objects as these are problem specific features. Of course, within a specific domain these can be subject to further discussion but the basic framework by which to undertake dynamic evaluation effectively needs to be established first.

In conclusion, the satisfactory solution of dynamic decision problems will necessitate the resolution, and considered avoidance, of several operational difficulties. These can be categorised as: design problems, e.g. confusion over problem terminology and unfamiliarity with decision theoretic techniques; problems arising as a consequence of opting for a dynamic treatment of the decision scenario, e.g. operational overheads incurred; and impediments to performance stemming form model constraints, e.g. restricted value interactions. Throughout the design process care must be taken to balance the demands of implementation against the need to ensure evaluation integrity is maintained.

.

Chapter 4

Approaches for Undertaking the Analysis of Multiple Attributes

If one considers a mobile ad hoc network in which a node changes location, has its role re-assigned or encounters different environmental conditions whilst in operation, then one can anticipate that under such circumstances allowing only statically defined behaviours will restrict both the node's, and consequently system's, ability to respond to the variations in conditions encountered. Whilst such networks may exhibit an extreme degree of dynamism, both in terms of the system's architecture and functionality, some portion of these experiences will not be knowable in advance. Aspects of design, such as distribution, data-centric behaviour and user-interaction all increase the level of uncertainty associated with activities undertaken on-line. This is one argument for supporting the ability of a system to undertake dynamic decision resolution. Also, most such decision scenarios will necessitate that more than one criterion is explored if the problem is to be investigated at a level acceptable to the decision maker.

There are numerous ways of undertaking multi-criteria analysis and it is reasonable to argue that the choice of appropriate method be given some consideration, not least because subsequent actions may critically depend upon the validity of the evaluation results. In the case of dynamic decision resolution an additional factor to be considered is the impact of the dynamism upon the act of evaluation itself. This may give rise to repeated instances of the problem being encountered, rather than undertaking a one-off evaluation. It may also generate subtle differences in the problem definition over time which will also need to be allowed for. Whilst only three particular approaches are discussed below the intention is to raise awareness of issues generally relevant to the discussion of dynamic multi-criteria evaluation. The choice of evaluation methods primarily arose as a consequence of reviewing existing decision theoretic techniques. These are by no means exclusive, nor are they suggested as predominantly the most satisfactory approaches, but they do collectively illustrate three different techniques, each with associated advantages and disadvantages. The three chosen types are:

- Weighted additive multi-attribute value functions (WAM)
- Enumeration and scoring method (ENUM)
- Lexicographical sorting method (LEX)

The universal problem of trading-off the ideal requirements against the practicalities and capabilities of an operation again surfaces here, specifically between the complexity of design and off-line analysis against the expressiveness and overheads of on-line operations.

All three methods present results which are in a single value form with respect to each specific alternative object. In the following, the term 'depth first' describes evaluation which proceeds by analysing all attributes for a single alternative before continuing to the next object whereas 'breadth first' describes the converse situation where all instances of an identical type of attribute are evaluated across all alternatives prior to the next attribute type being addressed.

4.1 Weighted Additive Value Model

4.1.1 Description

The weighted additive value model is a well-established form of multi-criteria analysis with a proven theoretical underpinning. Multi-criteria analysis (MCA) falls within the branch of research termed Decision Theory and, as such, is intent upon the provision of rigorous and meaningful techniques by which to resolve decision problems [Whi76]. Perhaps the most immediately well-known method in this area is that of Expected Utility Theory, as propounded by von Neumann and Morgenstern [Neu47], which applies to the evaluation of uncertain outcomes in the presence of risk. However, many other multi-criteria methodologies also exist including the Analytical Hierarchical Process and the Outranking Method [Vin92]. In addition, there are a variety of techniques, primarily drawn from the fields of economics and operational research, such as efficiency frontiers and multi-variable optimization methods which are also applicable in this area. Of all these many approaches, the weighted additive model was chosen for discussion for several reasons. Firstly, as a form of MCA, it continues the proposals made in [Pra98] to investigate this area of research with respect to finding solutions to the problem of providing dynamic values in a real-time context. Secondly, it directly represents notions of trade-off and preferential expression between attributes, two activities which are integral to the general discussion here. It also has the advantage of being readily understood and having a simplicity of operation which is not complicated by consideration of probabilities, as required for models dealing with uncertainty. More importantly, however, is the fact that it represents an extant theoretical model for exploring decision problems, where the criteria involved can address a variety of topics, and where a successful solution must consider issues of subjectivity on the part of the decision maker. Weighted functions, whether motivated by a full understanding of a theoretical model or derived arbitrarily, are a common approach for dealing with multi-attribute problems. In many respects the WAM method presents an ideal solution to the difficulty of choosing an evaluation methodology by which to resolve such multi-faceted decision scenarios and yet, in a dynamic context, it is not without some problematic operational issues.

Under the WAM approach, a group of alternative outcomes are characterised by a set of attributes which are deemed appropriate to the exploration of the particular problem. The individual state of each attribute is assessed by the application of a value function and a weighting constant. The resulting attribute values are then combined, in this case by summation, to give a collective value by which the overall relative attractiveness of the particular alternative can be identified. That is it provides a numeric indication of the ability of an alternative to satisfy the defined problem. The results of an evaluation will consist of a single value being assigned to each alternative object, which is indicative of its state with respect to the problem statement. This value can then be used to establish the attractiveness of alternatives within the group evaluated or to compare the alternatives against some global baseline. The collation of individual attribute values can, alternatively, take a multiplicative form, the relevant approach emerging as the problem is defined. Under WAM the preferences of the decision maker are explicitly examined during the definition stage but may not be readily apparent in the results. The appealing aspect of this, and other, decision analysis techniques is that they enable myriad disparate issues to be represented in a numeric form thus facilitating comparisons. A detailed account of this method is provided in [Kee76], the authors of which have carried out much work in this area.

The difficulty in applying an additive value model lies not with the finished form but in interrogating the decision maker in the first instance so as to reveal his value structure and relative preferences. It is necessary to capture these if it is going to ever be possible to identify a successful resolution of the problem. In the general case the decision maker is asked a series of questions with the intention of revealing his particular preferences for each individual attribute being queried. This process leads to the definition of value functions and weighting constants which inherently reflect the underlying preference structure. Thus, the numeric components emerge as a direct consequence of this investigation process, leading to the definition of a function of the form,

$$V_a = w_1^a v_1^a(x_1^a) + w_2^a v_2^a(x_2^a) + ... + w_n^a v_n^a(x_n^a)$$

where V_a is the overall value associated with alternative object *a* and w_i^a *and* v_i^a represent the weighting constant and value function, respectively, of attribute x_i. The flexibility of the method is illustrated in the literature by a range of scenarios from the mundane resolution of the problem of which domestic appliance to purchase to the siting of a reservoir. However, the size of a problem is not indicative of the existence of strong preferences and it may require careful questioning and repeated consideration of all issues involved until the decision maker is clear about his belief structure. It is also important to remember, when the additive value method is applied, that the results are only a valid representation of the underlying preference structure so long as the properties dictated by the model are adhered to.

For the multi-attribute value form, where the number of attributes exceeds two, the principal requirement is that attributes are *preferentially independent* that is for a pair of attributes (x_1, x_2) their preferential order is independent of the values of all remaining attributes. There are also additional independence conditions, particularly applying to the use of utility functions, all of which should be checked during the function definition process. The significance of maintaining preferential independence in the multi-attribute case is that it uses the fact that a pair of attributes is independent of all other attributes within the set of attributes being considered to facilitate the process of identifying relative preferences. Establishing the validity of value functions when only two attributes are employed requires a different approach involving establishing that the corresponding trade-offs condition is met [Kee76]. Therefore, whilst on the one hand this method provides an established model by which to handle the tricky problem of extracting and formalising the decision maker's preferential beliefs, it also carries with it certain obligations to correctly observe the properties associated with the model if the results produced are to remain a true interpretation of these beliefs. Also, because of these obligations, safeguards must exist against subsequent alterations to an evaluation once it has been defined which might undermine the representation of the original belief structure upon which it was constructed.

As Keeney and Raiffa point out [Kee76], it should also be understood that, in the WAM method, the weighting factors employed are scaling constants, indicative of the relative desirability of attributes over their scaled value range. They should not be misconstrued as a token of the attribute's absolute worth. For example, if considering the purchase of a car the question of safety might be deemed inherently more important than price. However, if all options present a similar restricted value range for the 'safety attribute' the weighting for that criterion will be lower than for the price attribute, where the value range may be very large and the difference more noticeable. Thus, the value functions and weights are dependent and each criterion cannot be treated in isolation of the other criteria employed in the evaluation. The danger, in this situation, is that the failure to observe the requirements of the model or to overlook the correct implementation may not prevent the production of a seemingly plausible result. In a real situation it may be impossible to detect whether such a result is an accurate reflection of the decision maker's preferences or, rather, a significant misrepresentation.

Operationally, the weighted additive model would be more suited to a 'depth first' approach to the interrogation of attributes, that is it evaluates each alternative object in its entirety before proceeding to the next.

4.1.2 Critique

The great strength of the WAM method is its ability to cope with an extended range of inputs in a consistent manner. So long as the constituent value functions can handle the raw attribute values and components of the evaluation do not vary then the weighted additive model can be relied upon to produce a valid result. This makes the method highly robust with respect to fluctuations in attribute state. Theoretically, it is also highly scalable in that no restriction is placed upon the number of attributes which can be defined or number of alternative objects which can be evaluated. In practise, however, it would seem reasonable to restrict the number of attributes on the basis that there is a diminishing return with respect to being able to distinctly resolve a problem when more attributes are defined by which to investigate it. Also, it increases the amount of analysis required of the problem in order to reach a functional definition.

A feature which can be construed as being of particular benefit, as far as this discussion is concerned, is that the WAM approach represents a systematic way of representing issues of preference and in a manner which allows for the theoretical justification of the consequent results. That is, the approach is able to capture and explore notions about the relative attractiveness of attributes. Indeed, this is the objective of the whole process as the value functions are not intended to reflect purely mechanical changes in the amount of an attribute. This reflects the fact that it is not feasible to directly compare measures of different units. Also, that straightforward assessments of quantity may not coincide with the importance of the attributes being quantified in relation to their ability to satisfy the overall selection problem.

If considering the adoption of WAM based evaluation as the basis for ongoing, dynamic and adaptable multi-criteria evaluation, certain limitations are perceived. The application of the multi-criteria analysis techniques mentioned earlier only ever addresses a single, specific instance of a decision problem and cannot be generalised to deal with several, slightly varying problems. Obviously, once a solution has been arrived at and applied to a static problem then it ceases to be a problem and if a problem still persisted then some aspect of the definition must have changed, meaning the scenario would need re-examining from the beginning. For dynamic and adaptable on-line evaluation scenarios, it will be the norm for the identical problem to re-occur, given the repetitive nature of system activities. Allowing for no change in components or belief structures, then the same value structure can be re-applied without difficulty, justifying the requirements for dynamism in evaluation. However, an adaptable evaluation implies that re-occurrences of the problem are not necessarily met with the identical response. The elements involved may be changeable and, as a consequence, also the relative desirability of the attributes. Indeed, even if the set of attributes did maintain the same relative value ordering, if, as a consequence of adaptation, fewer of them were analysed then the existing weighting constants too would also need to be re-evaluated. Alternatively, if the dynamic evaluation tolerates the possibility of there being a change in the problem specification, then an alternative definition and treatment will be required in order to identify the correct response for the new problem context as, without the full and proper examination of each problem scenario, the results cannot be claimed to be justifiable representations of the decision maker's belief structure. The problem, here, lies not so much with the active evaluation, although there might be delays in loading methods not recently accessed, but with the additional design workload required to define all the evaluation variants.

A second issue refers to the characterisation of attributes. As has been stated, it is a requirement of the model that attributes are preferentially independent of each other. This is unlikely to be the case in many real decision scenarios in a system context where many problems will involve issues of resources and functionality and where the elements by which the problem is represented are not independent. The modification of attribute values by the application of weights also inherently implies that attributes are tolerant of trade-offs between each other. Again this is an assumption which, in many instances, will not be justifiable. In general design problems the use of weighting constants is a common tool by which to apply favour to certain elements during problem analysis, or equally to penalise the less attractive options. The values employed are frequently arrived at in an ad hoc fashion with no analytical justification by which to back-up the choices made. Not only does this make the presence of weighted values theoretically problematic it also makes it difficult to define methods by which such weights can be identified and modified on-line. Within the context of the weighted additive model discussed here, there is a theoretical underpinning but it may not necessarily be appreciated by those implementing the evaluation and trial and error weighting values substituted instead.

Also, whilst it is argued that there is a need for dynamic evaluation scenarios to consider subjective, e.g. preferential and quality issues, these are not the sole factors to consider. Many decision problems contain a mix of subjective and objective issues. To this end, it should be noted that value functions as employed in multi-criteria analysis are not supposed to be direct mappings between purely physical measurements onto a representational scale, rather they are supposed to contain an element of subjectivity.

For simple selection scenarios defining a weighted additive evaluation may be seen to be unnecessarily complex in its approach when a solution can be easily arrived at by establishing the maximum of a particular attribute, for example. In addition, owing to the functional transformation of individual attribute values and the composite nature of an alternative object's state, it may not be easy to distinguish which factors have influenced particular outcomes, particularly if viewed retrospectively. This may be of concern during the verification of any subsequently upgraded process.

4.2 Enumeration and Scoring

Having considered the case of the WAM method which offers both a theoretically proven means to account for issues of preference and a succinct evaluation form, an alternative approach is now considered which is not burdened with theoretical constraints but rather opts for a more direct manner when handling a decision problem.

4.2.1 Description

Given a clearly defined problem and knowledge of what outcome will constitute problem satisfaction it would seem convenient to adopt an evaluation approach which takes as its input a statement as to what the desired outcomes are, for example $attribute\_a > 0$ and $attribute\_b \geq 10$, and compares the set of alternatives directly against this. This is the basic approach presented by the enumeration and scoring method defined here. It is motivated by the intention of providing a method of multi-criteria evaluation which is adequate for general purposes and easily defined, without the burden of requiring expert knowledge relating to its application.

However, even simplistic approaches, such as this, must still be applied in a methodical and consistent fashion and a description of the method follows.

Given the restricted nature of many system-based decision problems, particularly in the real-time domain, as discussed earlier, then their satisfaction can be readily established by comparing the set of available alternatives against bounding parameters which constrain performance. It is assumed that it is possible to order the criteria defined on the basis of importance, or preference as is the abstraction employed here, and then to award scores in accordance with the size of ordering achieved. By enumerating attributes and value levels of interest, and awarding scores to the alternatives on the basis of their ability to satisfy requirements, composite scores can be arrived at which can then be used to undertake a relative comparison of the alternative objects.

Unlike the previous WAM example, the ordering of criteria is assumed and not established as part of the evaluation process. The ordering does not have to be complete, there can be equivalencies between criteria and also dependencies. Conceptually, for each decision problem a *preference ordering, $order_i$,* is defined which comprises one or more *preference statements, s_j^i*. In turn, a preference statement contains one or more *groups of expressions*, $g_k^{\ j}$, where each expression interrogates some selection of the available criteria, together with any relevant constraints on those criteria. Each preference statement is associated with a particular score and all groups contained within a statement are deemed to be of equivalent worth and thus earn the same score if the expression is satisfied. Only a single expression within a statement need be satisfied as each expression is viewed as being independent of its peers, criteria which are dependent begin encapsulated within the same expression. Whilst each preference ordering is associated with only one selection problem, a selection problem may be associated with more than one preference ordering reflecting changes in conditions and preferential assumptions which arise as a consequence of these changes. Each ordering may include one default statement to account for situations when none of the preferred options are satisfied. By these means the evaluation can be enabled to be responsive to variations in its environment. In order to be effective, and meaningful, it is necessary that all statements and expressions are non-contradictory and resolvable.

The trade-off of attribute values can be expressed by way of different preference statements, as can 'threshold values' which represent the boundary point at which the preferential relation between two attributes changes. A defined preference order, therefore, assumes a structure akin to the following outline:

For decision problem, *a*,

$$
\begin{aligned}
&order_a \\
&\{ \\
&\qquad s_1{}^a = \{ g_1{}^1 \colon\ g_2{}^1 \colon \ldots,\ g_n{}^1 \};\ \ score\_s_1{}^a = w; \\
&\qquad s_2{}^a = \{ g_1{}^2 \colon\ g_2{}^2 \colon \ldots,\ g_n{}^2 \};\ \ score\_s_2{}^a = y; \\
&\qquad s \ldots \\
&\qquad s_n{}^a = \{ g_1{}^n \colon\ g_2{}^n \colon \ldots,\ g_3{}^n \};\ \ score\_s_n{}^a = z; \\
&\}
\end{aligned}
$$

where *1, 2* and *n* represent the 1^{st}, 2^{nd} and n^{th} positions respectively in the preferential ordering of statements and $g_i{}^n$ is the i^{th} group of expressions for the statement positioned at level *n*. The statement scores (*w, y, z)* monotonically increase or decrease with respect to statement order, depending upon the definition of the problem. Thus, for the case where $s_1{}^a$ represents the most preferred statement, the preferential relations are $s_1{}^a\ P\ s_2{}^a\ P\ s_n{}^a$, with $g_1{}^1\ I\ g_2{}^1\ I\ g_n{}^1$ and $g_x{}^1\ P\ g_x{}^2\ P\ g_x{}^n$. Identifying an appropriate ordering is either an ad hoc process relying on expert knowledge or may utilise the mechanism mentioned in Section 5.4 for establishing a preferential ordering. The scores can be generated for each level of statement by employing an order preserving function, in this case $v(s_r) = (1/2)^r$, which exhibits the property displayed in Equation [4.1],

$$v\,(s_r\,) > \sum_{j=r+1}^{n} (1/2)^{\,j} \qquad [4.1]$$

namely, that the sum of all lower order statements' scores cannot outweigh a single score of the next highest order statement. This resolves the problem of using an additive combinatorial form, where there is a danger that the satisfaction of a large number of lower ordered statements might outweigh a smaller number of satisfied higher order statements. As the preferential ordering contains no notion of magnitude then such behaviour cannot be theoretically justified. The rationale underlying the score assignment, however, is that ordering is a known factor and that it should therefore be preserved where possible in order to support consistency in interpretation of results. The general assumption is that preferences will be structured in such a way as to require that the maximisation of total values will indicate greatest preferential satisfaction.

Enumeration and scoring might be equally suitable to undertaking an investigation of the alternatives either depth or breadth first, with the appropriate choice being dependent upon the context of the decision problem and the target system.

4.2.2 Critique

Electing to investigate a decision problem by directly listing and scoring the various criteria in terms of their desirability, either with or without reference to real values, may have the appeal of being a transparent and quick way of achieving an implementation. The state of specific criteria, as represented by their values, can be specifically targeted and all lesser conditions ignored. However, this apparent simplicity, when compared to the previous WAM example, is misleading as the latter method addresses the question of both identifying the underlying preference structure and the production of a representational form, whereas here the existence of preferential orderings are taken as an assumption and the work required of an enumeration model is, therefore, reduced. The nature of the preferred attributes and critical values is explicitly stated, making revision of problems easier to achieve and allowing for customization to specific requirements. Equally, dynamic modification can be effected without raising concerns of issues of attribute dependencies, and variations in attribute types can be readily addressed. Operationally, partial results, if they have been derived consistently across all attributes, may still be of some use. Also, the incorporation of constraint values into the preference statements allows objects to be compared both with global baseline values and with each other, on the basis of their overall score.

Whilst restrictive model requirements may be bypassed, thereby reducing the off-line complexity, this has the effect of increasing the on-line difficulties. The granularity of the investigation, i.e. the range and quantum of values, is dependent upon the level of detail specified in the preference statements. Clearly, it is impracticable to score every possible value outcome even if they can be enumerated and, consequently, scoring is more suited to serving the identification of categories of objects and checking the lower bounds on performance. As the number of expressions and scoring levels employed in the ordering increases then structuring of statements can become difficult and the potential for instigating an excessive number of operations, which will be an impediment to performance, will also grow. Whilst the limitations on expression, made by requirements of consistency, may be few, under the approach employed handling multiple attribute and value sets can quickly lead to a growth in possible alternative states during the handling of which, it is easy to deviate from the stated preference model.

Another difficulty arising from the supposed advantages of being able to conduct evaluation based upon comparisons with constraining parameters is that it then becomes necessary to provide precise definitions for these and know what exactly the constraining values are and what conditions their occurrence relates to. This activity may require some ‘fine tuning’ via simulation or monitoring of real situations. Theoretically, an individual score can be associated with the undertaking of any computational method, such as maximisation, minimisation or application of some complex function. Whilst self-defined scoring schemes allow for the customized treatments of attributes, the risk of excessive overheads being incurred through their usage should not be overlooked.

4.3 Lexicographic Criteria Analysis

As with the weighted additive model, lexicographic analysis is also a known approach to resolving multi-attribute decision problems [Fis74]. If the previous case of enumeration/scoring proved less theoretically complex than its predecessor, then the lexicographic variant on evaluation is the least complex of the three, which is its primary attraction.

4.3.1 Description

Using a lexicographic approach to decision resolution, the criteria, designated X_i by which the problem is being examined are ordered in terms of their worth to the decision maker. A lexicographic ordering is one where, for alternatives a' and a'', then $a' \, P \, a''$ if and only if either,

$$X_1(a') > X_1(a'')$$

or

$$X_i(a') = X_i(a''),\ i = 1, \ldots, k \qquad \text{and} \qquad X_{k+1}(a') > X_{k+1}(a''),\ \text{for some } k = 1, \ldots, n-1$$

meaning that if a' returns a larger score for criteria X_1 it will be preferred to alternative a'' [Kee76]. Again, establishing the preferred ordering of attributes is not considered directly within the method but rather is an assumed precursor to the method being applied. Evaluation commences with the most important attribute, which is assessed across all alternatives. If, after this first round no clear winner has been identified then the second most important attribute is considered and so on until a clear winner emerges, at which point evaluation terminates. Once eliminated from a round, an alternative cannot be again considered for subsequent rounds and no notion of the cumulative state of an attribute is maintained between rounds. It is owing to the abrupt way in which attributes are dismissed when employing lexicographic reasoning and the fact that decisions can be effected on the basis of, in the minimum case, only one criterion having been examined, that the technique is disapproved of as a rigorous decision making tool [Kee76].

This method dictates a ‘breadth first’ approach, the evaluator traversing all instances of an identical type of attribute. There is no requirement to maintain a running total of individual object values, only the identity of those objects still remaining in the evaluation. Also, in contrast with both the WAM and ENUM methods, under the LEX method, the value assigned to an alternative object is directly dependent upon the comparison of one or more of the object's attributes with those of the other objects. Therefore, for an efficient operation all other alternatives should be known and available otherwise the evaluation process must either wait until they become visible or drop alternatives which are delayed. Any comparison with a global value would theoretically need to be done as part of an individual attribute's appraisal.

4.3.2 Critique

In contrast to WAM, a lexicographic approach values a speedy resolution of a decision problem over a thorough resolution. As Payne et al note, with reference to human decision making strategies, approaching the problem on an attribute-first basis may be more effective when there are time constraints in evidence,

> *"when deciding in high-velocity environments, the decision maker should focus on breadth of evaluation rather than on depth of evaluation. That is ..... considering all the multiple options on important attributes"* [Pay96]

and this, together with its ‘administrative ease’ [Kee76], may make it worthy of consideration regardless of failings in other areas. Indeed for scenarios where a single issue is dominant then this approach may be highly suitable. The method is inherently conservative in its operation with evaluation terminating as soon as a winner is found, thereby producing a useable result within the minimum of operations. The method is also intuitively straightforward, overlooking the requirement to order the attributes in the first instance, and maintains a clarity of actions which makes it more readily understood by external observers than the other evaluation methods. A significant difference from the previous two cases, however, is that the results are conceptually of a binary nature with "preferred/not preferred" semantics. There is no necessity in the model to maintain any reference to the real attribute values and all states relating to those alternatives which fail the evaluation are discarded. As with the WAM method, there is still a requirement that attributes be independent of each other in order to maintain the simplicity of the mechanism. If this were found not to be achievable, then attributes might be redefined and the level of interrogation shifted to a higher level of abstraction.

The primary defect when employing a lexicographic approach to the interrogation of alternative options has already been referred to, namely its incomplete coverage of all the defined criteria. This raises the question of whether, when there is no guarantee that an attribute will ever be employed during a dynamic evaluation, there is any merit in investing effort in their initial identification and definition. The effect of this abrupt behaviour needs to be considered in the context of undertaking repeated evaluations, where, over the course of time, evaluation could become predominantly a single-issue matter, with the attributes of lower importance never being taken into account. Alternatively, the least preferred attribute could find itself continually employed in the role of 'tie-breaker' which brings into question the original choice of higher order attributes. When defining evaluation scenarios for time constrained environments it should be noted that the 'most important' attribute may not, in reality, be the 'most decisive'. No matter how significant the contribution an attribute may make towards the resolution of a decision problem, there is little advantage to the resolution of that problem if all alternatives persistently exhibit the same value for that attribute.

In all three of the methods which have been discussed the results are, of course, dependent upon attribute states in order to generate a tie between attributes and, hence, alternatives. However, under the LEX method, as each attribute is considered in isolation there is no accumulated difference maintained between them by which to reduce the possibility of such a draw occurring. Any equivalence in the final attribute will result in a failure to identify a single winning alternative regardless of the fact that, for all previous attributes the alternatives may in reality have achieved high, but matching, values. The isolation in which attributes are handled also precludes any notions of trade-off between them.

4.4 Comparative Analysis

4.4.1 Rationale

Chapter 5 will consider factors involved in designing and implementing a dynamic multi-criteria evaluation in general, such that the selected form of evaluation methodology is appropriate to the nature of the problem in hand. In order to undertake this process effectively the properties and limitations of any potential evaluation methodologies must be realised. The objective, therefore, in undertaking a comparative analysis is partly to reveal the individual characteristics of each of the sample evaluation methods. However, the context of this discussion is one of dynamic and adaptive evaluation processes, where selection scenarios are repeatedly encountered and where conditions may change between each instance. Therefore, there are additional factors to be considered, namely will the chosen evaluation form tolerate on-line modifications and what will be the effect of repeated evaluation upon system resources. The outcome of this analysis can then be used as input to any subsequent design activity.

In view of the previous statements, a general issue is one of the flexibility of an evaluation methodology, particularly as relates to two aspects of its behaviour:

- The expressiveness of the method in being able to deal with requirements as desired, given the nature of the selection problem.
- The ability to support the dynamic treatment of the problem via runtime modifications.

In exploring these issues both on- and off-line activities need to be considered as achieving a satisfactory realisation of an evaluation scenario will be influenced by the trade-off made between the design and runtime resources invested in the problem. To a large extent, the burden of enabling dynamic evaluation falls not with the on-line portion of activities but rather with the off-line portion where multiple definitions and decision scenarios have to be explored.

The particular aspects identified as worthy of further consideration are listed below. There is a basic division of activities into primarily on- or off-line concerns.

Expressiveness:
This is restricted to meaning the ability of the evaluation methodologies to deal with both subjective and objective issues (as defined elsewhere). The capability to investigate a variety of specific topics is limited only by the knowledge of the designers and physical properties of the intended system; there is no constraint on what can be defined as an attribute or on how an attribute's state can be measured as such. When discussing generic evaluation, however, one must allow that definitions as to what constitutes a satisfactory resolution to a decision problem may reference a mix of individual preferences, pre-defined constraints and comparative evaluations of basic physical properties. Expressiveness is considered to be an off-line concern.

Subjectivity is incorporated into evaluation by enabling certain attributes to be ‘favoured’ over others, such that their contribution to the result is in excess of what an equivalent proportion would be. Subjectivity, in the abstract, represents a property which cannot be derived from knowledge of the current system. That is, it has its origins in time and location outside the target system's scope. Therefore, subjective values must be input directly into the evaluation by the decision maker. These can be motivated by a variety of factors e.g. cost and experience, and here are termed as ‘preferences’. Subjectivity will emerge initially in the choice of attributes to interrogate but within an evaluation function itself it is most commonly represented by the inclusion of a weighting constant dedicated to enhance, or reduce, a particular attribute's value. In addition to the use of weights, the other principle means of including decision maker's subjectivity is through the translation of raw attribute measures via value and utility functions, which are defined to be relative to a particular set of conditions. A third way of accommodating subjectivity into operations is to increase the amount of attention paid to a component such that it receives a differentiated level of service, for example an object can be allocated more than its fair share of physical resources or allowed greater time in which to complete its work.

Taking the above three techniques into account, it becomes apparent that the weighted additive model, with its use of weights and value, or utility in the case of risk scenarios, functions can readily include a subjective component. It might also be structured such that key attributes were appraised ahead of less significant ones if their state was to be intercepted before completion of the evaluation to serve as a partial indicator of conditions. The enumerated/scoring method allows for subjectivity by employing an ordering on attributes and the scores they achieve. Value functions could be defined on an attribute but this might undermine the requirement for a scored output as such. Likewise, the lexicographic approach also uses ordering as a means to subjectivity and, again, value functions might be employed, though in this case weights would have no role as attributes are considered in isolation.

Problem Size:

Taken from the view of the design perspective rather than on-line operations, this has a direct bearing on the amount of analysis and off-line preparation which must be undertaken in order to define and construct the evaluation proper. Focussing on defining the problem rather than handling a specific number of objects, for example, if it takes x units of work to specify one attribute on average then electing to explore ten attributes might justifiably be assumed to increase the work required proportionally. Indeed, additional attributes could lead to an exponential growth in workload depending upon the evaluation methodologies used. There is also the question of whether the chosen evaluation methodologies are familiar to the designers and the principles upon which they operate fully understood. Lack of awareness of such issues will cause an impediment to the progress and accuracy of an implementation.

In terms of the three examples used here, the weighted additive value model is the most theoretically rigorous and intensive as regards the demands it makes in relation to interrogation of the decision maker. The complexity associated with the enumerated approach is focussed upon checking consistency of expressions and implementation issues. The lexicographic approach can be quickly assimilated off-line, though this is taken in isolation of establishing the preferential orderings which here is a separate process. The size of the evaluation problem is an issue which is closely related to that of operation complexity.

Complexity:

Of critical importance to any system are the runtime overheads incurred by performing an operation. In order to maintain independence of implementations and provide early indications of what such overheads might be algorithmic complexity can be substituted in advance of obtaining monitored values so as to provide an approximation of the length of time execution will require. There is also the question of the space required by the evaluation during its execution for handling the specified data types and to what degree values are ephemeral, stored or, indeed, duplicated between objects and evaluator. Being prescriptive about the complexity of a particular evaluation method is not possible unless it comprises known algorithms.

The three cases discussed here are presented as heuristics rather than defined algorithms as the complexity of any implemented evaluation methodology will be to a large degree dependent upon the worst-case attribute appraisal method employed and this will be specific to the nature of the attribute itself. Another difficulty is that, complexity analysis does not consider the overheads relating to the location of activities and/or objects. Again, for an evaluation process employed within a distributed context, the overheads of communication will outweigh local processing costs.

The complexity of the three examples used here will be discussed subsequently.

Scalability:

The issue of scalability can be separated into two components: having the ability to handle an increased number of attributes and having the ability to handle an increased number of objects. The former is an off-line, design problem, as attributes cannot be created in an ad hoc fashion once a process is implemented. The latter is an on-line characteristic of the evaluation problem which may raise operational problems but will have no impact upon the evaluation methodology as such whose structure and operation should be independent of the anticipated number of alternatives made available. To this end, the three methods are indifferent to the number of objects handled.

Modifiability:

Implicit in a discussion on dynamic evaluation is the notion that any methodology should be flexible enough in its structure that it can tolerate some form of adaptation. This might, alternatively, be perceived as customisation. This idea is captured by determining how it would respond to modification, a topic which will be discussed further in Chapter 5. Three specific ways of carrying out modification on-line are considered, i.e. by reducing the number of attributes being interrogated, by substituting an attribute type or by substituting the form of attribute appraisal. The second and third variants are designated largely as off-line problems as their identification is integral to the definition of the evaluation. Once defined, however, varying the number of attributes in response to fluctuations in conditions can only be carried out if it does not undermine the validity of the evaluation's results. Some methodologies will be more tolerant of such fluctuation than others.

Robustness:

This is akin to the previous situation but differs slightly in that it considers whether the methodology is capable of producing a partial result which is still viable or whether it operates in such a way that a termination of activity prior to its natural completion is likely to prevent a result, in its incomplete state, being of any use. The treatment of attributes or objects which are temporarily omitted, owing to being unavailable for some reason, is implementation dependent but they theoretically present no impediment to evaluation continuing, for example they might be set to generate a zero value in such an instance. Questions of robustness are of relevance to distributed environments where conditions are highly dynamic.

4.4.2 Evaluation

In order to investigate the above points an evaluation was performed based upon a hypothetical problem scenario of identifying a location to spend a month's holiday in the UK, where each location was characterised by weather and its proximity to the sea. Whilst this represents a scenario dealing with a single evaluation instance, rather than an ongoing dynamic problem, it is necessary to establish the inherent individual characteristics of each methodology before proceeding to investigate issues of dynamism further and is, therefore, justified. Four attributes were defined to represent rainfall, A_0, sunshine, A_1, temperature, A_2, and distance from the sea, A_3. The values for the first three were taken directly from data sets produced by the Meteorological Office [MET], giving average monthly and yearly figures for the years between 1914 and 2005. Attributes a_0 and a_3 are in reality negatively defined, that is the lower their value the more desirable, so their value was inverted so as to allow a positive definition akin to attributes a_1 and a_2. The data sets were fixed across each evaluation variant in any one test in order to facilitate comparisons.

A fundamental obstacle in this instance is being able to define precisely equivalent specifications for each of the three forms of evaluation discussed, owing to their differing capabilities. For the purposes of defining scaling constants for the weighted additive value model, the attributes were preferentially ordered $a_0 \ P \ a_3 \ P \ a_2 \ P \ a_1$ and given values of 0.4, 0.35, 0.2 and 0.05 respectively. For the enumerated and lexicographic variants the interpretation of preferential ordering is subtly different; under the ENUM method preference is effectively expressed for attributes and value points, whereas under the LEX approach the predominant preferential expression is for the attribute alone. This difference in expression will exhibit itself in variations in results, as will be shown. Under the WAM method linear value functions were applied to each attribute's measured value and these were retained for the other two approaches, largely for convenience.

The three approaches discussed here as possibilities for performing multi-criteria evaluation dynamically have so far been described in an abstract manner and it is expected that an actual implementation may be realised in any number of ways. For the purpose of evaluation, here, they were defined in a straightforward fashion as separate methods by which a number of ‘alternative objects’ were evaluated. The general structure takes the form of two loops to locate the appropriate attribute and an inner block to perform the work of extracting and processing the attribute value and then assigning this before proceeding to consider the next value. The LEX method differs slightly in its structure from the other two, as shown below in the pseudo-code descriptions of each method's functionality:

WAM

outer loop:	identify attribute
inner loop:	identify object
body work:	obtain attribute raw value; apply value function; apply weight; update object value
end loops:	End

ENUM

outer loop:	identify attribute
inner loop:	identify object
body work:	obtain attribute raw value: apply value function; check score [series of comparative tests for each attribute]; update object score;
end loops:	End

LEX

outer loop:	identify attribute
inner loop:	identify object
body work:	obtain attribute raw value; apply value function; compare current attribute value with current leading value [if greater, update new leading value & record id; if equal - record tied value]
end inner loop:	compare tied value with leading value [if equal proceed to next attribute; if not tied value < leading value; assign winning object flag; end]
end outer loop:	End

For WAM and ENUM, the outer and inner loop conditions are swopped to account for either the breadth first (outer loop on attributes) or the depth first (outer loop on objects) approaches. For LEX, the outer loop condition referenced attributes only as this is the inherently logical approach to follow. For ENUM and LEX, although it is not required to employ value functions as such they were retained for reasons of consistency and convenience. This was not considered to affect the analysis of results as their presence can be viewed as a marker for some form of further computation in a real situation. The methods were instrumented to count the number of operations employed, approximately as described above, including value assignments, with the number of operations counted being used in the following to provide an indication of the time complexity involved. Note that operations are only counted up to, and including, the assignment of the final object values. Issues of sorting and usage were designated functions of the consuming process of the evaluation, not aspects of the evaluation itself.

There is an argument to be made for excluding the action of attribute measurement from an analysis of the overall evaluation method if the architecture was such that it was independently executed by a process external to the evaluating process, as befits, for example, a scenario with highly autonomous objects. Here, however, it is included for completeness.

The worst-case, *wc*, performance for each of the three methods as described above, is given below, where *n* is the size of the outer loop, *k* the size of the inner loop and *w* the worst-case number of operations in the body:

WAM: $wc = (nkw) + n$

ENUM: $wc = ((nk2) + (xw)) + n$ $\quad$ x = number of alternative objects

For the lexicographic approach, *n* represents the number of attributes (outer loop) and *k* the number of objects.

LEX: $wc = 2nk + (6(k-1) + 2)n - 1 + 6k + 4n + 7$

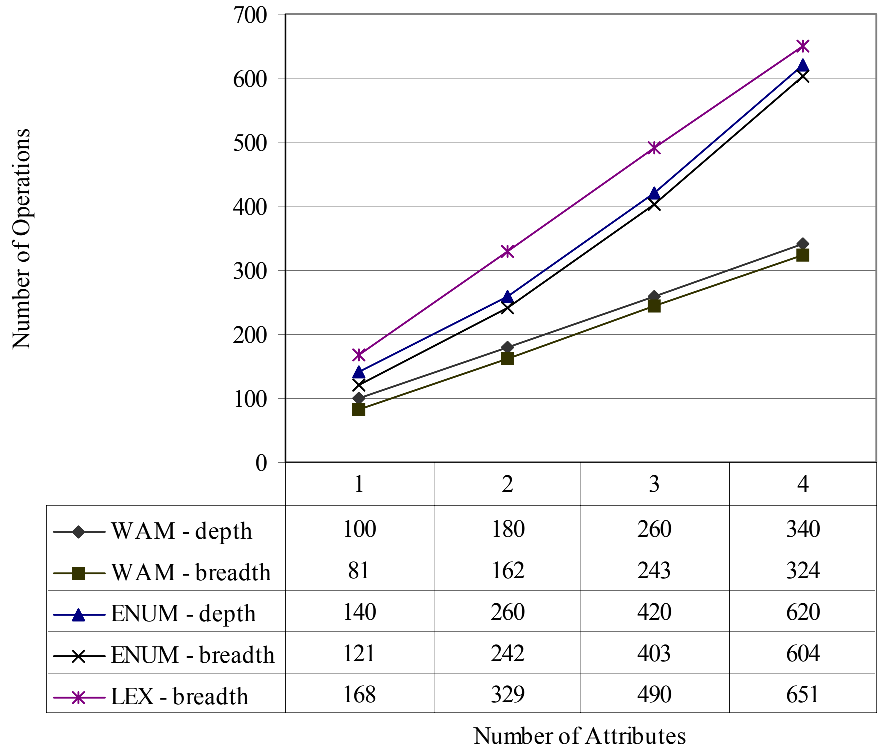
For the ENUM method, as represented in this implementation, the worst-case occurs when every attribute encountered on all objects presents the minimum value required to score, thus ensuring that the maximum amount of comparative actions have been performed. For the lexicographic method, the worst-case reflects the scenario where, for each attribute excluding the lowest ordered one, each of its measurements across the set of objects is better than all previous ones with the exception of the last measurement which equals the last but one, thus giving a tie, and for the lowest ordered attribute the final measurement does not tie, but is better than all the others.

Although the specified values are implementation dependent they do illustrate that all three have a complexity of $O(n)$ which is to be expected when each attribute, and hence object, must receive identical treatment if the results of evaluation are to be consistent. There is a fundamental difference between the operation of the weighted additive model and the remaining two approaches. In the former, the performance is independent of the real values held by attributes and thus constant. For the lexicographic and enumerated approaches, however, performance is heavily influenced by the nature of attribute values, both in terms of their magnitude and their distribution. This makes WAM highly predictable in its operation but inherently indifferent to the state of attributes within a specific problem. Under ENUM and LEX, however, whilst the two methods are heavily based on comparative operations and thus give more pessimistic worst-case performance levels, actual performance in a real situation would be dependant upon their particular specification and the actual attribute values encountered. Therefore, in some circumstances, they might be capable of generating results more quickly than if WAM were employed.

Chart 4.1 shows the number of operations generated by each approach as the number of attributes is increased. For this example the data set was adjusted to generate the worst-case conditions as described above. The scaling constants for the WAM method were as given earlier and for the lexicographic approach the preferential ordering of attributes was a_0, a_1, a_2, a_3. Twenty alternatives were explored in all cases. These parameters were fixed for both the depth and breadth first approaches.

Using the same worst-case data set and four attributes, as before, the impact on performance of varying the number of objects was tested. The results are shown in Chart 4.2. The number of objects considered commenced at one and was then incremented by one each step, up to and including, 20 objects. The slight 'kink' under the transition from one to two objects under the LEX method is unavoidable as, when considering only one object there are no comparisons to make, meaning no possibilities of ties and the object, therefore, goes through after the first attribute has been tested. In all other cases, the evaluation encounters ties and runs through to the last attribute. For both charts, the slight improvement in performance, with respect to the number of operations, of the breadth first approach over the depth first variant is due to the structuring of the methods.

To illustrate typical, rather than worst-case, behaviour the evaluation was performed using the meteorological data set mentioned previously. Twenty alternatives and four attributes were explored, using values arbitrarily selected from the data set which remained the same under each method. For the WAM method, the scaling constants used remain unchanged at 0.4, 0.05,



	1	2	3	4
WAM - depth	100	180	260	340
WAM - breadth	81	162	243	324
ENUM - depth	140	260	420	620
ENUM - breadth	121	242	403	604
LEX - breadth	168	329	490	651

Chart 4.1: Effect of Varying the Number of Attributes on the Number of Operations under Worst-Case Conditions for the Three Evaluation Methods

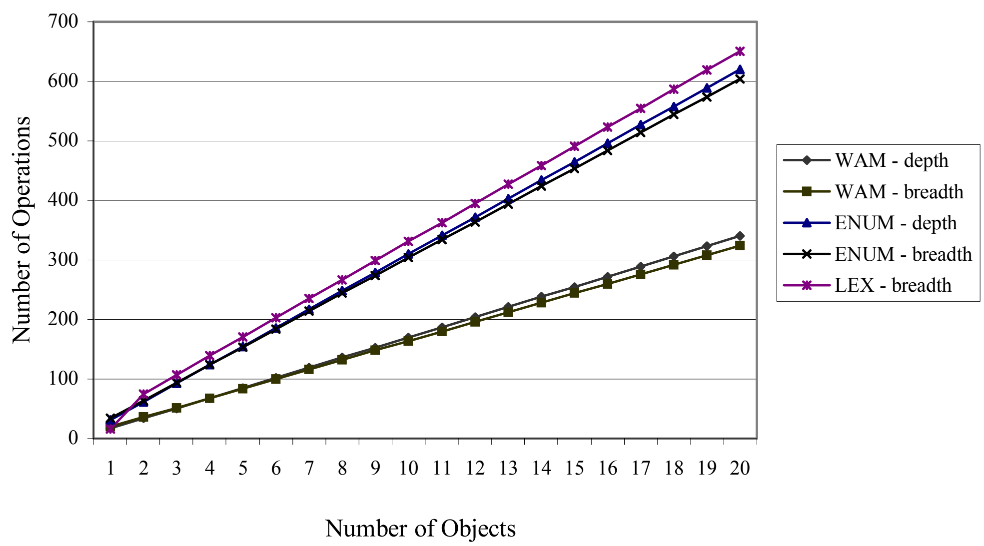


Chart 4.2: Effect of Varying the Number of Objects on the Number of Operations under Worst-Case Conditions for the Three Evaluation Methods

0.2, 0.35 for a_0, a_1, a_2, a_3 respectively. For the LEX approach the preferential ordering of attributes was a_0, a_3, a_2, a_1 which was intended to be interpreted as meaning that it is most important that the chances of rain are minimised, then that the sea is nearby, then that it was reasonably warm and, finally, that the decision maker is not overly bothered about the amount of sunshine.

Under the enumerated approach the situation required further definition. The notional preferential ordering of attributes, reflecting how significant it was that their values matched ideal requirements, was the same as under the lexicographic example, a_0, a_3, a_2, a_1. However, on further consideration it was felt that, as the rainfall figure was presented as an average value, it could only be taken as an approximate indicator of conditions which might be encountered on the ground, so to speak. That is, less than desirable levels of rain could not be ruled out as extreme sample points. Therefore, it was pre-empted, in terms of the amount of attention paid to its state/value set, by a_3, the attribute giving the distance from the sea, as these values represented specific distances and, therefore, the level of satisfaction they engendered could be counted as certain. The level of interest in each attribute type was reflected in the number of comparative conditions checked. For a_3 three checkpoints were defined, giving three possible levels of scoring which might be achieved, for a_0 and a_2 two levels were defined, whilst for a_1 only one level was given. This produced, in total, eight different scoring levels, the scores and the real-world conditions they reflect are shown in Table 4.1. As can be seen, the scores were not contiguously assigned, rather they were allocated according to the perceived desirability of that particular condition being met. For example, the most desirable condition was identified as being that rainfall was less than 25 mm per month, whilst the least significant condition, to merit identification that is, was deemed to be that the distance to the sea was between 5 and 25 miles. The maximum score achievable by an alternative object was 0.7891.

Attribute	*Assigned Score (score - position)*	*Corresponding Attribute Value*	*Real Condition*
Rainfall (a_0)	0.5 - 1	0.875	25 mm rain per month
	0.0625 - 4	0.8	40 mm per month
Sunshine (a_1)	0.00781 - 7	0.5	150 hours per month total
Temperature (a_2)	0.03125 - 5	0.84	21^{o}C mean
	0.01563 - 6	0.64	16^{o}C mean
Distance from Sea (a_3)	0.25 - 2	0.995	1 mile
	0.125 - 3	0.975	5 miles
	0.00391 - 8	0.875	25 miles

Table 4.1: Value Checkpoints and Scores Employed in Enumerated Method under Typical Behaviour (Chart 4.3)

At the termination of the weighted value and enumerated methods, each alternative will have been assigned a value indicating how well it has performed under this particular problem, which will enable the consumer of the evaluation results to interpret their relative status whilst still having some indication of their individual state. The lexicographic approach, however, only identifies a single winner, as a winner rather than giving its actual value, deterring any subsequent attempt at ranking of the alternatives. Ranking can be achieved under the lexicographic approach, but it requires the repeated application of the method to the set of alternatives, with the removal of the leading alternative from the set at the end of each evaluation cycle. The results of applying this repetition are shown in Table 4.5.

Three separate runs were made using three different sets of data and the number of operations generated as a consequence of each run is shown in Chart 4.3. Under the repeated lexicographic approach, which is not shown in Chart 4.3, the total number of operations executed was: *January* - 880, *June* - 916 and *December* - 1622. The increased workload under the third data set occurred because only in that set was it necessary to consider the second, third and fourth attributes on occasion before a winner emerged. In the first two runs evaluation terminated after the most significant attribute was considered, i.e. at the end of the first round, as a single leading alternative had been identified. Ties emerge as a consequence of alternatives rising in significance as more valued alternatives are removed. In terms of consideration of the issues attributes represent, the WAM method addresses all attributes, the ENUM method addresses

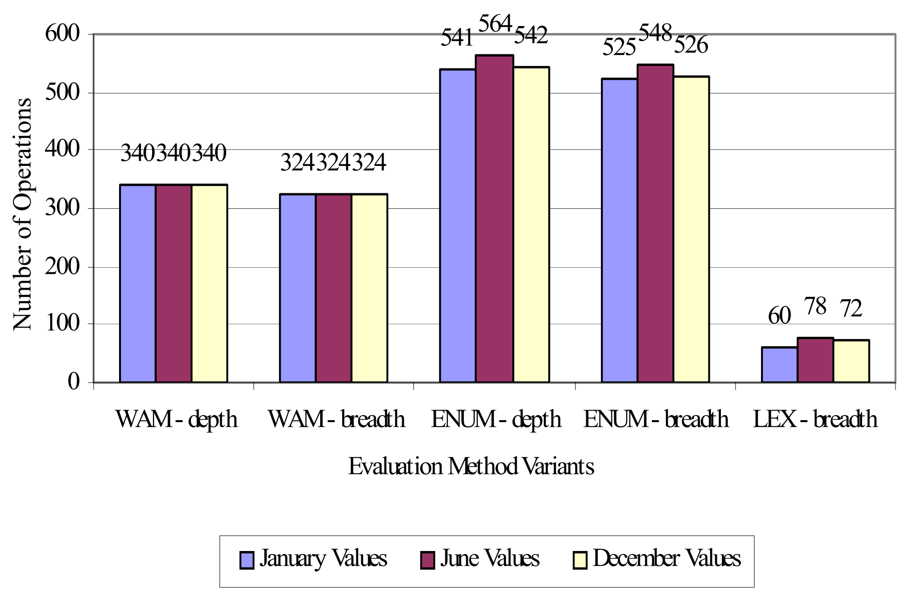


Chart 4.3: Number of Operations Generated by Evaluation Method Variants for Three Separate Data Sets

them precisely as specified, which in this case was all, and the LEX method addresses only as many as it has to in order to produce a result. With the ENUM method, certain alternatives may achieve no score, as shown, owing to the nature of the expressions.

Tables 4.2 - 4.4 show the evaluated order imposed upon the set of alternatives as a consequence of applying each variant. The differences in results between each method, when using the same data set, arise from the subtle differences inherent in the way the preferences have been expressed and serve to show that notions of predictability must be broadened when handling subjective issues.

Under the WAM method, there were no equivalencies within the final values, allowing a complete ordering of alternatives. Likewise, when the LEX method was repeatedly applied to the alternatives. Under the ENUM method, however, as Table 4.3 shows, there was a tendency for values to be grouped owing to the coarse granularity of expressions used. The number of discrete categories which it is possible to identify within the set of alternative objects will be a consequence of the scoring combinations, including the case where no score is achieved. If the number of alternatives being examined is greater than this amount then it is inevitable that some duplication of objects' scores will occur. The probability of equivalencies occurring under all three methods is influenced, not only by the number of alternatives, but also by the individual attribute type's value range and size.

Position	***1st***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20th***
Jan	7	1	6	19	18	4	17	14	3	8	2	9	10	5	13	15	12	16	20	11
June	14	2	6	4	1	9	15	19	12	13	11	17	18	8	7	10	20	5	3	16
Dec	15	4	19	3	11	8	13	7	14	18	5	17	20	2	9	12	10	16	6	1

Table 4.2: Order of Alternatives under WAM Method by ID (Identical for both variants)

Position	***1st***	***1***	***1***	***2***	***2***	***2***	***2***	***2***	***2***	***3***	***3***	***3***	***3***	***4****	***4****	***4****	***4****	***4****	***4****	***4th***
Jan	8	15	20	1	6	7	14	18	19	2	4	9	17	3	5	10	11	12	13	16
Position	***1st***	***2***	***3***	***3***	***3***	***4***	***4***	***4***	***4***	***4***	***5***	***5***	***6***	***7***	***7***	***8***	***8***	***8***	***8***	***8th***
June	14	12	8	15	20	1	6	7	18	19	2	4	13	9	17	3	5	10	11	16
Position	***1st***	***1***	***1***	***2***	***2***	***2***	***2***	***2***	***2***	***3***	***4***	***4***	***4***	***4***	***5****	***5****	***5****	***5****	***5****	***5th***
Dec	8	15	20	1	6	7	14	18	19	5	2	4	9	17	3	10	11	12	13	16

* This category achieved no score

Table 4.3: Order of Alternatives under ENUM Method by ID (Identical for both variants)

Pos	*1st*	
Jan	7	Remaining alternatives all fail
June	14	
Dec	5	

Table 4.4: Order of Alternatives under LEX Method by ID

Position	***1st***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20th***
Jan	7	1	5	12	3	19	17	6	13	16	18	4	14	10	8	9	2	15	11	20
June	14	12	13	4	2	9	6	1	5	10	15	11	16	19	17	18	20	8	3	7
Dec	5	3	20	6	14	1	18	8	15	4	13	11	16	12	19	7	9	17	2	10

Table 4.5: Order of Alternatives under Repeated Application of LEX Method by ID

An illustration of the variability in results produced under different evaluation methods is provided in Chart 4.4 which presents the orderings for the June data set in graphical form. This shows how the impact of expressing concern for specific attribute conditions and differences in preferential expression can cause fluctuations in an alternative's relative position.

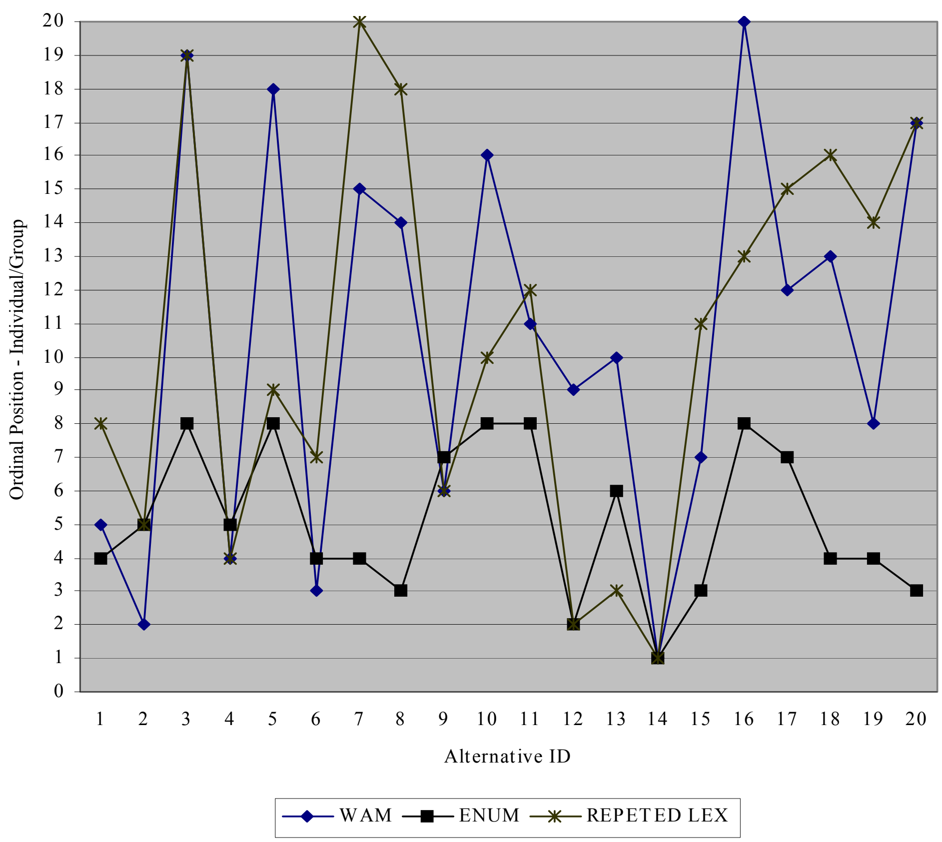


Chart 4.4: Ordering of Alternatives under Different Evaluation Methods (June Data Only)

Equivalent attribute, and hence overall, values are indicative of a large degree of similarity in conditions existing between alternative objects. In the case of the LEX method, for example, whilst the possibility of all attributes producing the worst-case conditions might never be ruled out, if this state were to be persistently encountered it would call into question either the need for an evaluation in the first instance or the choice of attributes employed.

An aspect of behaviour which is of particular interest is how variations in available computation time will impact upon the quality of the evaluation and the usability of any results produced. There are two approaches which can be taken when exploring this issue: the first is to view the problem in the abstract and inspect the state of evaluation if the process is arbitrarily interrupted which, in a real scenario, would require the continuous update and notification of values to the consuming process in order to ensure that a partial result might be available; the second course is to assume that an evaluation is modifiable, has knowledge of the time available to it and has adjusted its behaviour accordingly. The first tactic was realised by generating a random value, to fall within the worst-case performance range, and assigning it to each evaluation variant so that progress, in the form of number of operations accrued, could be checked repeatedly against it. At the point at which the number of operations were equalled or superseded by the control, the process terminated and a record was taken of the number of attributes and objects handled up to and including that point. The outcomes of runs under typical and worst-case behaviour are presented in Table 4.6 and 4.7.

It can be seen from these snapshots on progress that the WAM method has gone further through the alternative objects and attributes than the other two methods in all cases. The LEX method has finished in the typical scenario, not because it is inherently quicker, but because the data was such that a winner was found after analysing the first attribute. The design implication arising from Tables 4.6 and 4.7, is that there are three options when it comes to the production of results arising from the partial evaluation of all objects and attributes. The focus can be to obtain:

- Completely evaluated objects - favoured by depth first approaches.
- Completed attributes across all objects - as produced by breadth first approaches.
- The maximum number of attributes completed as a proportion of objects - as produced by the breadth first approach by considering the results of the incomplete round of evaluation which was ongoing at the point of termination, i.e. for the final termination value entry, 278, for the WAM breadth case, instead of taking 20 objects with only 3 attributes assessed, i.e. the last complete case prior to failure, the alternative would be to consider only 8 objects but each of which had been addressed in full for all 4 attributes.

Termination Value	*Evaluation Variant*	*Objects - Depth First Approach*		*Attributes - Breadth First Approach*		*Current Processing Position at Termination [object - attribute]*
		Number of Objects Evaluated in Full	*% of Completed Objects*	*Number of Attribute Types Evaluated in Full*	*% of Completed Attribute Types*	
76	WAM - depth	4 (100%)*	20%	-	-	5th obj - 2nd att
	WAM - breadth	-	-	0	0	19th obj - 1st att
	ENUM - depth	2	10%	-	-	3rd obj - 4th att
	ENUM - breadth	-	-	0	-	16th obj - 1st att
	LEX - breadth	-	-	1	25%	
222	WAM - depth	13 (62%)*	65%	-	-	14th obj - <1st att
	WAM - breadth	-	-	2	50%	15th obj - 3rd att
	ENUM - depth	8		-	-	9th obj - 2nd att
	ENUM - breadth	-	-	2	50%	3rd obj - 3rd att
	LEX - breadth	-	-	1	25%	
316	WAM - depth	18 (63%)*	90%	-	-	19th obj - 3rd att
	WAM - breadth	-	-	3	75%	18th obj - 4th att
	ENUM - depth	11	-	-	-	12nd obj - 4th att
	ENUM - breadth	-	-	2	50%	17th obj - 3rd att
	LEX - breadth	-	-	1	25%	

* The difference in % of objects completed between WAM depth and ENUM depth.

Table 4.6: Amount of Evaluation Completed under the Typical Data Set given Random Termination

Termination Value	*Evaluation Variant*	*Objects - Depth First Approach*		*Attributes - Breadth First Approach*		*Current Processing Position at Termination [object - attribute]*
		Number of Objects Evaluated in Full	*% of Completed Objects*	*Number of Attribute Types Evaluated in Full*	*% of Completed Attribute Types*	
138	WAM - depth	8 (100%)*	40%	-	-	9th obj - 1st att
	WAM - breadth	-	-	1	25%	14th obj - 2nd att
	ENUM - depth	4	20%	-	-	5th obj - 3rd att
	ENUM - breadth	-	-	1	25%	3rd obj - 2nd att
	LEX - breadth	-	-	0	0	17th obj - 1st att
595	WAM - depth	20 (5%)*	100%	-	-	-
	WAM - breadth	-	-	4	100%	-
	ENUM - depth	19	95%	-	-	20th obj - 2nd att
	ENUM - breadth	-	-	3	75%	20th obj - 4th att
	LEX - breadth	-	-	3	75%	14th obj - 4th att
278	WAM - depth	16 (77%)*	80%	-	-	17th obj - 2nd att
	WAM - breadth	-	-	3	75%	9th obj - 4th att
	ENUM - depth	9	45%	-	-	10th obj - <1st att
	ENUM - breadth	-	-	2	50%	5th obj - 3rd att
	LEX - breadth	-	-	1	25%	15th obj - 2nd att

* The difference in % of objects completed between WAM depth and ENUM depth.

Table 4.7: Amount of Evaluation Completed under the Worst-Case Data Set given Random Termination

In order to explore the second tactic of modifiability, the nature of such modifiability has to be considered. Substitution has already been identified as a topic which needs to be primarily addressed off-line and so the focus is on the dynamic adjustment to the number of attributes employed in an evaluation. An illustration of the effect such modification might have on the number of operations required in the worst-case can be seen in Chart 4.1 but it is also of interest to see the difference such a reduction might make on the ordering of alternatives as this will give an insight into the consequences of employing this particular approach. Charts 4.5 - 4.7 show the effect upon evaluation results by the removal of a_3 (distance to sea) from the evaluation, using the June data only. Attribute 3 was chosen as it did not dominate the evaluation but was still of sufficient significance for its omission to make an observable difference. In order to accommodate the change in the evaluation structure it was necessary to assume that the weighted value model was reassessed so as to provide alternative scaling constants. These were reset to 0.51, 0.167, and 0.323 for a_0, a_1 and a_2 respectively. In a genuine situation, there are two issues relating to such a process of re-definition which may impede progress: depending upon the nature of the evaluation problem being discussed re-appraisal might be a far from trivial process; secondly, unless the decision maker's subjective values for each attribute are strongly distinct, then any definition may, in reality, carry insufficient meaning to justify the process. The attribute ordering for the LEX method was accordingly adjusted to a_0, a_2 , a_1.

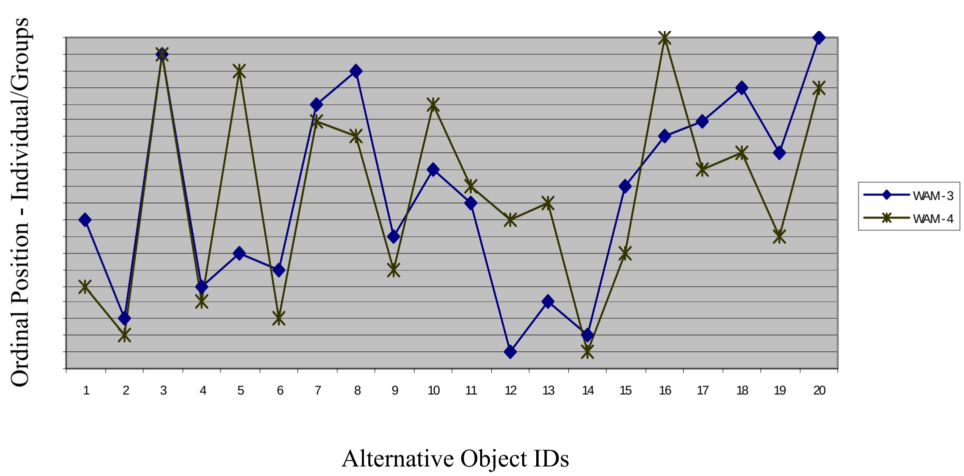


Chart 4.5: Impact of Removal of Attribute 3 on Alternatives' Positions - WAM June only

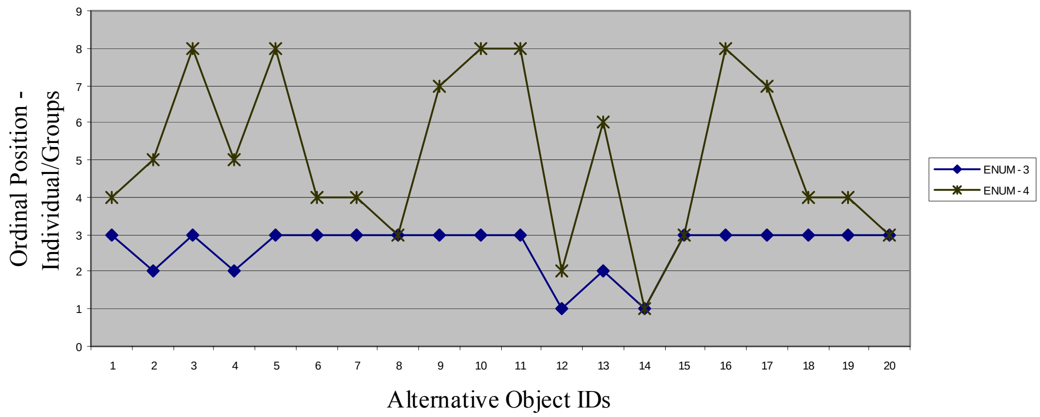


Chart 4.6: Impact of Removal of Attribute 3 on Evaluation of Alternatives - ENUM June only

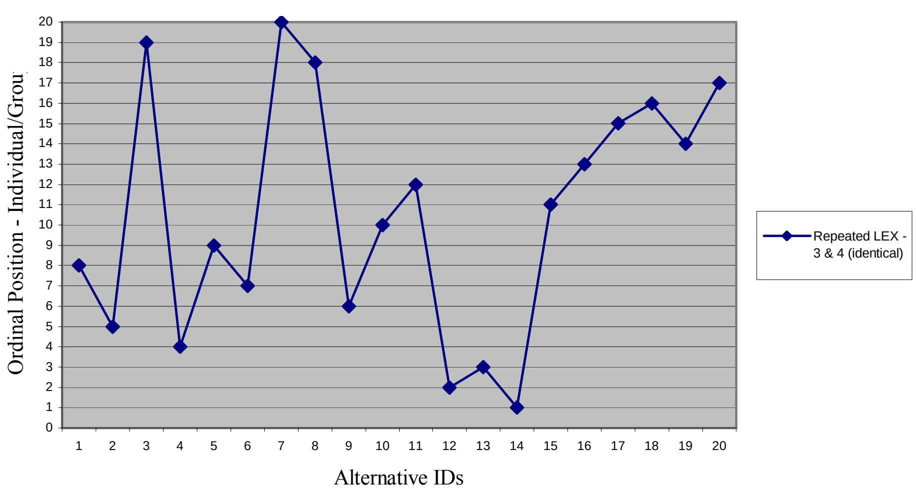


Chart 4.7: Impact of Removing Attribute 3 on Alternatives' Positions - Repeated LEX June only

Under the LEX method the advantages of employing attribute dropping as a means of modification are somewhat limited as it may seldom be necessary to analyse more than one attribute anyway. To illustrate how minor might be the role played by lower ordered attributes under the LEX method, the repeated LEX approach was employed, using 20 objects and 4 attributes, with a different data set on 36 occasions, making the theoretical total number of ties possible 2736 (assuming one object is removed each round). In reality, there were found to be only: 14 instances of both the first and second ordered attributes being used; 7 instances of the first, second and third ordered attributes being used; and 5 instances where all four attributes were used. In all the remaining cases inspecting the first attribute alone was sufficient to identify a winner.

When the reduced evaluation was compared with the full, 4 attribute approach, the consequences for each of the sample methods was quite different. Under the WAM method there is a degree of similarity in the results, with a few substantive differences. The ENUM method exhibited a much reduced range of categories arising from an equivalent reduction in the range of values. The scoring conditions had been set quite high (refer to Table 4.1), to reflect requirements in line with a satisfactory resolution of the problem, and many alternatives where only able to score anything via the use of a_3. Consequently, when this was removed, their scores were reduced. Also, the other attributes did not have the advantage of having three scoring levels associated with them, so there was less opportunity to differentiate in cases where attribute levels were greater than the lower bound. Indeed, when the reduced evaluation was run on the January figures for ENUM, no scores were achieved at all, indicating that the results exhibited in Table 4.3 for January are based entirely upon scores associated with a_3. This may initially appear incorrect behaviour but is, in fact, a precise interpretation of the stated meteorological requirements - nowhere in January was able to produce a match to the type of conditions wanted. The two sets of results were identical under the LEX method. This was to be expected as, under this particular data set evaluation of a_0, the leading attribute, produced a winner after the first round and a_3 would not have been reached in either case.

The implications for employing this form of dynamic modification are that in all cases the variation between full evaluation results and the reduced evaluation will differ greatly depending upon the significance of the dropped attribute's role. For WAM this might be expected to be less as the values are a composite of all attribute states and the re-adjusted weights may smooth out extreme tendencies. For the other two approaches, however, it is possible for one attribute alone to exert a great influence on the outcome and, if this is removed, then the consequent results will be significantly different.

Regardless of the issue of modification, the previous observations have some bearing upon the question of specifying multi-criteria evaluation procedures in general, particularly in relation to knowing what attributes to define and at what point to limit the number of attributes being handled. The inclusion or exclusion of an attribute will change the results in a manner which does not necessarily mean that they are 'better' or 'worse' but only different. Likewise, the belief that adding more attributes will enable a greater differentiation in results and thus facilitate more clearly delineated operations, may not be justified.

4.4.3 Discussion

There are four inter-related areas of concern relevant to the question of employing dynamic multi-attribute evaluation within a system. These are the issues of:

- Dynamism - e.g. Is the evaluation responsive enough to cope with fluctuations in the subject matter?
- Expressiveness - e.g. Is it possible to cover the range of issues in the manner desired?
- Design/Engineering - e.g. What is the workload associated with development?
- Operational - e.g. What are the overheads associated with running the evaluation?

These factors should prompt certain questions to be asked of the proposed problem definition and the intended form of its resolution. Examining potential evaluation methodologies will provide solutions to some of these probable questions. From the analysis of these particular three methods, their various characteristics might be summarised as follows:

- WAM - suited to complex stable evaluations where attribute independence can be guaranteed and attributes are tolerant of trade-off. Can be used for both local and global evaluation scenarios. Thought required incorporating constraints into operations.

- ENUM - suited to highly changeable, low complexity problems where attribute/state dependencies may exist. May be particularly useful for exploration and trial of preference-driven problems to test expectations on outcome against stated requirements. Also for scenarios where it is sufficient to identify lower bounds on performance are met. Can be used for both local and global evaluation scenarios. Readily supports expression of constraining parameters and tolerant of their being changed.

- LEX - suited to the speedy resolution of an abnormal evaluation problem, e.g. single issue eviction. Also applicable to memory constrained environments owing to low data retention. Equally, it might suit evaluations where there genuinely is only one dominant issue and lesser attributes are assigned precisely to resolve any tie-break situations. Best suited for local evaluation scenarios.

However, a more structured recording of the observations generated by the analysis might be necessary for future reference. This could take any one of a myriad of formats depending upon requirements, a straightforward variant being provided in Table 4.8. The observations entered here take one of three forms: ordinal as in 1(st), 2(nd), 3(rd); Y (yes), N (no), C (conditional); and numeric values derived directly from the previous analysis [value], relevant to the depth or breadth approach. The conditional entries would be supported by annotations, which have been omitted here. Of course, even when endeavouring to be objective, if the problem is not deemed sufficiently significant to warrant a formal program of analysis being defined then the interpretations derived risk being skewed by the personal experiences and knowledge shown by the evaluator of the methodologies. For example, notions of the difficulty associated with re-definition of a problem will depend upon how familiar one is with the definition procedure involved.

As a consequence of undertaking this exercise, not only will the decision maker/engineer's awareness of operational issues relating to the intended dynamic evaluation scenario have been raised, thus hopefully reducing the possibility of errors being made later, but he may also have gained a positive insight into the nature of the problem being considered, for example what level of examination, in terms of number of attributes and expressiveness, is appropriate.

4.5 Summary

This chapter has undertaken an examination of three possible techniques by which multi-attribute decision problems might be investigated on-line. It has illustrated some aspects of their operation and provided suggestions as to the type of questions which should be asked when contemplating the use of such methods within the context of the evaluation of a particular decision problem. It has also, hopefully, raised awareness of some of the problematic issues in this area and shown how the choice of methodology and nature of the method's definition influences the results produced.

One conclusion arising from the study, which is particularly relevant if they dynamic and on-going nature of evaluation scenario is remembered, is that it would be advantageous from the perspective of both off-line and on-line operations to match the type of evaluation method to the nature of the decision problem. For example, if the on-line decision problem is likely to exhibit

Area of Concern	*Issue*	*WAM (d/b)*	*ENUM (d/b)*	*LEX (b)*
Dynamism	Ease of enabling method to modify treatment of attributes	3	2	1
	Method suitable to attribute dropping	C	Y	N
	Method suitable to substitution of attribute appraisal form/type	C	Y	Y
	Scalable with respect to number of alternatives	1	3	2
	Method likely to benefit from modifying treatment of attributes (via impact on duration of evaluation)	2	1	3
Expressiveness	Ability to accommodate known constraints on attributes directly	C	Y	C
	Ability to accommodate notions of attribute trade-offs	Y	C	N
	Ability to describe dependencies between attributes	N	Y	N
	Any particular restrictions on number of attributes which can be accommodated	N	C	N
Design/ Engineering	Value structure identified as part of method definition	Y	N	N
	Least amount of theoretical underpinning/knowledge required of underpinning	3	2	1
	Suitable for employing global comparison	Y	Y	N
Operational	Least difference between number of operations in typical and worst-cases	1	2	3
	Least fluctuation in runtime overheads of non-adaptive method	1	2	3
	Least fluctuation in memory footprint at runtime	1	2	3
	For typical case: average number of operations involved in generating result	[340/324]	[549/533]	[70]
	For typical case: average number of attributes involved in identification of winner	[4/4]	[4/4] C	[1]
	Size of overheads dependent on values of attributes	N	Y	Y
Sample size: 20 alternatives & 4 attributes				

Table 4.8: Summary of the Characteristics of the Three Evaluation Methods

a large variation in number of alternative objects being appraised whilst the problem description itself remains unchanged, then of the three methods explored the WAM method would be the most appropriate choice, owing to its consistent runtime overheads and scalability. In contrast, if there were few alternative objects but the problem description and preferential structure were to be more variable, then a scoring mechanism such as the ENUM approach might be more suitable as, although it is less efficient when dealing with a large number of alternatives, it is more amenable to on-line modification of the criteria and comparative values and re-structuring of the scoring scheme. On the other hand, if the decision problem is concerned more with the speedy acceptance or elimination of alternative objects rather than a deep interrogation of several criteria, the LEX mechanism might by the optional choice of methodology.

This chapter has also shown that from the design perspective there is an option to place the emphasis upon either off-line or on-line activities. For example, the off-line choice might be to adopt a rigorous decision theoretic model including the careful elicitation of the decision maker's preferences as to the various criteria being examined. Such treatment, however, may make it difficult to accommodate on-line adjustments to the problem specification e.g. when conditions change or certain attributes are temporarily unavailable. In contrast, customised scoring mechanisms may increase the amount of on-line workload experienced per evaluation, but are more easily defined in the first instance and more amenable to adjustments once operational. Indeed, one possible advantage in the expressive flexibility of schemes such as ENUM is that it may make them a better choice if wanting to explore the impact of changes in the decision maker's belief structure, together with changes in any constraining values present, upon the level of satisfaction with evaluation results.

Finally, by examining the same problem under different evaluation methodologies, it has been shown that the expected results under each method may vary and, therefore, the treatment subsequently received by the evaluated alternative objects will vary too. This difference, whilst it arises from the choice of evaluation method, is a consequence in the subtle differences in the expressive capacity of each method and in the representation of the decision maker's preferences for the attributes involved.

Chapter 5

Analysing Dynamic Evaluation

There are two reasons why thought should be given to the choice of evaluation method. Firstly, there are the accepted goals of maintaining operational efficiency, avoiding redundancy of actions and safeguarding interaction with other system components. In addition, there is the need to ensure that the requirements of the decision making process can be provided for in terms of the expressive capacity and capabilities of the chosen evaluation method. There is also the related issue, which the possibility of dynamic evaluation gives rise to, specifically can a modified evaluation be supported, and is it reasonable to do so, without undermining the validity of results? Care has to be exercised in this respect as extending operations, by making them adaptable for example, may resolve one set of problems but only at the cost of introducing a new set of difficulties.

The following guidance is intended to enable the principle issues relating to dynamic multi-criteria evaluation to be discussed at a high level of abstraction and in a structured manner. The objective is to promote a clearer understanding of the fundamental nature of the proposed evaluation scenario and the degree of flexibility associated with it. Analysis allows for greater insight into the problem but, as with any design process, there are certain trade-offs to be made with respect to the various dimensions involved such as off-line design time, on-line overheads and number of attributes to be evaluated. The resolution of these can only be arrived at through context specific decisions so it is advisable not to be too descriptive at the outset. It is also difficult to consider factors which are implementation dependent with any degree of accuracy prior to any simulation or implementation occurring. Here, the emphasis is on facilitating consistency and adequacy in the employment of multi-criteria evaluation and in such a manner that it exhibits robustness in the face of fluctuations in the resource provision it receives. A methodical approach by which to address the twin aspects of appropriateness and adaptability of dynamic multi-criteria evaluation is presented.

A core assumption made here when discussing adaptive evaluation is that the non-adaptable, or ‘full’, evaluation is to be considered the normal behaviour and that any modifications applied which bring about a reduction in the depth of the investigation, whilst they may be necessary and tolerable, are less desirable. Thus, where conditions allow, the intention should be to effect evaluation in full and only resort to reduced evaluation under conditions of stress, for example as when resources are over-subscribed.

Three core operations may be required of any method by which dynamic evaluation is effected. The first is, obviously, that it be capable of handling multiple attributes. In addition, if relative differences are perceived by the decision maker in the contribution and worth of each attribute, then the chosen method will also need to exhibit a mechanism for differentiating between their relative merits. Finally, depending upon the type of evaluation, it may be necessary to ensure mechanisms within the chosen evaluation methodology which can account for constraints when arriving at a solution.

Analysis, as it relates to the definition and undertaking of a dynamic evaluation process, can be divided into two separate stages: those activities which are undertaken off-line during the design phase to reason about the definition and scope of the evaluation, and those activities which are undertaken on-line to reason about whether dynamic modification is required at any point in time.

5.1 Terminology of Evaluation

Certain terms and abstractions are employed in the description of the subsequent analysis, which are outlined here.

5.1.1 Definition of Terms

There is a general requirement to ensure that all participants in the design process have a common understanding of the terms which are to be employed in the ensuing discussions, particularly if they are ostensibly familiar words which are liable to context specific interpretations. It is also advantageous to be able to identify a generic set of role players who might potentially be involved in the evaluation process as these will help formulate functional decomposition as the design progresses. Section 3.2 introduced possible elements which might participate in the definition and operation of a dynamic evaluation mechanism, of which the following four positions are of immediate interest:

Decision Maker: The source of the value structure which motivates the form of the evaluation and what constitutes a satisfactory solution. In a system context the occupants of this role may differ from those who actually execute the evaluation, i.e. a software process. The assumption here is that decision making is a *human*, not machine, activity. That is, the dynamic resolution of decision problems requires input from a human factor in terms of what to evaluate, preferential expressions, and interpretations of worth. There can be a single occupant of this role or multiple occupants, as in the case of [Cla99].

Evaluator: As it implies, the identity of the object/process within which the evaluation process is embedded and which undertakes to perform the evaluation on-line.

Attribute Update Process(es): This is identified as a separate role in order to facilitate the assignment of functions at a later stage, particularly if the evaluation is to be employed in a distributed context.

Consuming Process: The process which consumes the results of an evaluation activity and whose requirements may influence the structure of the evaluation problem.

Of the other roles mentioned in Section 3.2, attributes and alternative objects will subsequently receive considerable attention and are more passive in nature so do not need to be singled out at this point. The data repository and adaptation manager are largely concerned with the adaptive nature of an evaluation and, therefore, will only come into play once a requirement for flexible support has been established.

It is not to be expected that the decision maker is automatically responsible for specifying the problem as well. The identification of a potential problem scenario might be made at another level in the design process by system engineers who have insufficient interest or knowledge of the specifics to act as decision makers themselves.

There is also the category of terms to consider which are specific to the problem such as geographical locations, component identities and the like.

5.1.2 Validation Check

In order to make dynamic evaluation worthwhile it has to bring about a result which would otherwise not have been achieved as an incidental consequence of another set of activities or which could not be produced off-line. Equally, any result which is achieved following analysis of a problem should be an improvement over that produced by a random selection process.

Two situations illustrate this issue. Firstly, at the network level packets may be serviced on a first come, first served basis and buffered until their turn to be forwarded. If the buffer overflows then packets will be dropped, thereby indirectly improving the service delivered to the remaining packets by reducing the level of congestion. Introducing a selection process with the intention of choosing packets to drop must, when all overheads are taken into account, produce a better level of performance than that achieved via random dropping. If not, then the behaviour which active selection introduces should be of more importance to the selecting process than the extra overheads incurred. This can be summarised as,

Benefit to Decision Maker = Benefit with evaluation - Benefit without evaluation

where the benefit to the system, however defined, should assume a positive interpretation. A second example considers that of fixed-priority task scheduling in a real-time system. Allocation of task priorities is a ranking procedure which, if it follows one of the proven algorithms such as the Rate Monotonic scheme, results in an optimal ordering upon the task set which indirectly ensures satisfaction of other properties. Adherence to deadlines is the dominant objective of the ranking process and no other properties can compensate for a failed deadline. If they do then it must be acknowledged that the dominant objective has been changed. From this arises the observation that,

> *If an operation is currently optimised to satisfy a critical constraint whose breach results in the failure of the operation, then no alternative optimisation can improve on performance whilst the critical constraint remains unchanged.*

If other factors are introduced to motivate behaviour, and constraints relaxed in the process, then it should be appreciated that the specifications have also been altered.

5.1.3 Evaluation Objective

This is a clearly stated definition of the objective which motivates the evaluation and in which the scope and key factors of the evaluation are named. It is vital that the objective and bounds of an evaluation are clearly identified not least because it makes it impossible to know when a satisfactory solution has been encountered otherwise. The more detailed and precise the definition is made, then the easier is the process of checking for compliance. The necessity of maintaining a trail of reasoning in an operational scenario is one motivation for undertaking a detailed definition process and a similar approach features in [Kon95a] though the focus there is confined to defining the evaluation criteria.

A single objective addresses one specific decision scenario. If the context changes, such that it gives rise to a change in objective or the belief structure which shapes the evaluation, then a new objective must be defined. In this way a highly dynamic decision problem can be composed of separate elements by employing multiple objectives. Objectives can also be structured hierarchically allowing several evaluation processes to be combined. In these cases, each objective reflects variations in the tolerances and the importance in the perceived state of attributes and objects involved.

An evaluation objective is characterised by several criteria which are summarised in Tables 5.1 - 5.3. For a static decision problem with a single instance of evaluation any such criteria would be concerned primarily with ensuring the accurate definition of that problem. However, for dynamic on-line decision scenarios there is also the operational aspect of the definition to be considered too. Hence, of the criteria listed in Table 5.1, with the exceptions of items i, vi, ix and x (objective definition, evaluation type, alternative types and attribute types) which are common to both static and dynamic decision scenarios, the remaining criteria are of relevance to on-line operational aspects. Item v, evaluation frequency, in particular deals with the issue of the dynamism which any on-line evaluation exhibits. For on-line evaluation problems operational aspects must be explicitly considered concomitant with those dealing with the decision theoretic issues as it must be assured that the consuming process is capable of handling the evaluation's results at all times.

Satisfaction (Table 5.1, item ii) must contain reference to the number of alternatives which are required to satisfy the evaluation process together with any minimum global value, if necessary. The response may be a mix of natural language and numeric terms.

	Criteria	*Description*	*Options*
i.	Definition	Description of objective and scope of problem.	
ii.	Satisfaction	A definition/description of what constitutes satisfaction of the problem.	
iii.	Size of Problem	Specifically, the maximum number of objects which are to be investigated, together with the maximum number of attributes, in suitable descriptive or numeric format.	
iv.	Criticality	The importance of the evaluation process itself to the overall functionality of the consuming process. If essential to process functionality, then an evaluation is termed *critical*, whilst if desirable, but a process does not fail if it is omitted, an evaluation is non-critical.	critical non-critical
v.	Frequency	The rate at which a new evaluation is instantiated. The general categorisation allows for options of single point, intermittent and continuous/repeat evaluation, with time specific sub-divisions presented in any particular scheme.	once intermittent repeat
vi.	Type	The underlying purpose of the evaluation which is either to rank the set of objects or to identify an appropriate sub-set. The type of evaluation references the basic categorisation of choice problems.	ranking reduction (*n*) categorisation
vii.	Comparative Form	Is a solution obtainable with knowledge of the alternatives alone or does it require additional comparative information?	local global
viii.	Value Uniqueness	Dependent upon the value distribution of constituent attributes and precision of chosen representation format. Generic options are given but can be replaced by more detailed statistical interpretations if available.	low medium high
ix.	Alternative Types	Defines the nature of alternative objects. See Table 5.2 for sub-criteria.	
x.	Attributes	Definition of the set of attributes by which to appraise the objective. See Table 5.3 for sub-criteria	
xi.	Constraints	Is the performance of an evaluation likely to be subject to restrictions, either in terms of its own operation, or as constraints it must account for in arriving at a solution, or a combination of both.	Level 1 Level 2 Level 3 Level 4

Table 5.1: Evaluation Objective Criteria

	Criteria	*Description*	*Options*
i.	General Description	A general description of the type of alternative object which is the ultimate subject of evaluation, e.g. characteristic properties, location etc.	
ii.	Individual Alternative ID	To identify instances of alternatives - one entry per alternative.	
iii.	ID-specific description	Particular details relating to a known object.	

Table 5.2: Alternative Objects' Criteria

	Criteria	*Description*	*Options*
i.	ID	Unique identifier.	
ii.	Definition	Description of the type of attribute, including the issue it addresses and justification for choice.	
iii.	Owner	Description of object/location to which attribute attached.	
iv.	Criticality	The significance of the attribute's contribution to the investigation and resolution of a problem.	mandatory partially-mandatory optional
v.	State Appraisal	This is the specific form by which an individual attribute type is assessed and a value produced by which its state is measured. Initially, it may only be possible to outline the nature of this measure until the specific evaluation method which is to be employed is identified.	
vi.	Trade-off Tolerance	Is the attribute tolerant/intolerant of being traded-off against other attributes?	tolerant intolerant
vii.	Value Dependency	As distinct from preferential dependency: is the attribute's 'worth' dependent upon knowledge of some or all other values?	dependent independent

Table 5.3: Attribute Criteria

The significance of the role played by an evaluation in the overall functionality of the consuming process can be represented by the notion of its *criticality* (Table 5.1, item iv). If it is essential to perform some level of evaluation in order to determine which path to take next, then the evaluation can be termed *critical*. However, if failure to carry out an evaluation does not prevent the consuming process from proceeding then it is termed *non-critical.* This categorisation can reflect the differences in the fundamental reason for undertaking an evaluation, that is between the case where there is a base-level behaviour which it would be desirable to improve upon via employing an evaluation of the current conditions and the case where it is not possible to define an acceptable base-level behaviour, resulting in the necessity to examine the available options dynamically. It should be noted that the description of criticality applies to the execution of an evaluation, not to the failure to produce a satisfactory result. That is, a scenario where the failure to identify a node with sufficient resources to support a task and a scenario where a web browser which is not the user's favourite tool is selected, may be considered to represent the extremes of critical failure but, ignoring the possibility of software or logical error, such "failures" are a direct consequence of electing to use dynamic evaluation and result from the accurate measurement and interpretation of attribute state, in accordance with the given definitions. Again, if an evaluation is required to assess a set of alternatives in order that they can be ranked, and they all return the same value, this does not indicate a failure of the evaluation process. An inconvenient result may still be a correct result and should be foreseen amongst the range of outcomes. An additional point, in relation to questions of failure, is that in dealing with subjective issues and systems which involve human end-users, definitions of failure may be difficult to refine and the probability of failure difficult to establish. Taking the browser example, on some occasions a user's dissatisfaction may cause them to abort the downloading process whereas on others they may have a more urgent need to view a web page and thus be more tolerant. Being identified as a critical evaluation does not necessarily negate the possibility of also being a modifiable one too.

The frequency of evaluation (Table 5.1, item v) has an impact upon the form of modification as an infrequent process may be less amenable to adaptation by skipping an instance. A single case of evaluation is considered to be sufficiently isolated to be a one-off occurrence, as for example where an exception might be raised, whereas an intermittent evaluation is one that conforms to an aperiodic arrival pattern. The semantics of a repeat evaluation are that it is reiterated as the problem which it addresses is encountered again and again. If real and exact frequencies are known then they can be substituted for the generalisations.

The type of evaluation (Table 5.1, item vi) refers to the fundamental objective of the decision problem as outlined in Chapter 3. The precise number of ‘winning’ alternatives required of a satisfactory solution may be included.

The comparative form (Table 5.1, item vii) deals with the question of how much knowledge is required of the other alternatives when attempting to evaluate a single alternative. The *local* option describes the situation where, in order to assess the relative attractiveness of an individual object, it is necessary to know the values associated with all other available alternatives. It is not possible to establish, however, whether the absolute maximum or minimum of the value range has been encountered. In contrast, a *global* objective implies the existence of an absolute, known baseline of values against which each alternative is assessed thereby allowing for the worth of an individual object to be identified in isolation of the knowledge of the states of any other peer objects. The existence of a set of absolute values implies that the minimum and maximum values obtainable under a given evaluation process are also known. An informal interpretation of the consequences of this categorisation is that, when values are local, the structure of the evaluation is such that a solution can be more frequently found as it is a case of using the 'best' of the available alternatives. In the case of a global evaluation, even though there is at least one alternative available, it might fail to achieve a sufficiently high standard to meet requirements. Local evaluation can cause difficulties in some scenarios, though, specifically if the full compliment of alternatives is not available for analysis at the same instance, as the 'best' value of the current group of alternatives can be superseded by late arrivals. If the consuming process is geared towards taking the highest value available, then this can lead to a high level of change in the selected option. In the context of value-based scheduling this is known as the 'admission problem' [Dav95][Ber01] and can be resolved by relaxing requirements for needing to employ the best available alternative, employing statistical techniques to assess values or tolerating the high level of disruption. The comparative property influences the amount of information required as input into the evaluation process, either at design time or on-line, i.e. in the form of baseline values.

The uniqueness of a value (Table 5.1, item viii) is, of course, dependent upon the combined states of the individual attributes. This, together with the precision of the value and the physical properties of the types used in an implementation, is of relevance as it constrains the distribution of results. For example, if the largest value is represented by a floating-point data type, reduced to five significant figures, and which ranges between .00000 and .99999, then the result can be any one of 100,000 numbers. Whilst the actual measurement is dependent upon the nature of the attribute itself, the probability of a tie under such circumstances might be expected to be lower than if the value took the form of an integer data type and was restricted to just one significant figure. Conversely, does the subject matter of evaluation really justify the production of high precision values when obtaining the results may bring about additional overheads? A value of 0.11119 is greater than 0.11115 but, given the circumstances of their particular evaluation scenario, is it substantially better and really worth differentiating behaviours on? In short, for a set of alternative objects employing the same attributes, if the values generated only tolerate a low precision then it is reasonable to expect more ties between objects emerging in the results than if the granularity of the value is high. In turn, this factor will have a bearing upon the production of usable results, i.e. if there is a requirement that a clear winner be identified at the conclusion of each evaluation cycle then if the probability of ties being produced is significant an appropriate handling operation needs to be provided or, more correctly, the choice of attributes reconsidered as the primary objective is to distinguish between alternatives, not identify their similarities. Consequently, this issue needs to be considered partly because it may indicate that there is an inappropriate choice of attributes and partly as it promotes consideration of how to handle occasions where there may be no clear winner.

The focus of constraints (Table 5.1, item xi) within an evaluation can be used to further characterise behaviours. Four situations can be identified, as listed below in order of the increasing level of restrictions they make upon the operation of an evaluation. These are evaluation problems:

- Level 1 - with no constraints
- Level 2 - with constraints applying to nature of solution
- Level 3 - with constraints applying to nature of evaluation
- Level 4 - with constraints applying to both nature of solution and evaluation

It may be that Levels 2, 3 and 4 could be encountered, in any combination, at any time, on-line. Level 1 is deemed mutually exclusive with the remaining three and is to be interpreted as meaning that 'no constraints will ever apply'. Such a situation would allow for a thorough and complete exploration of a problem, with unlimited time and unlimited investigation of attributes in the extreme situation. In practise, it would permit the decision maker a free hand in defining the evaluation. If Level 2 is expected to apply, then the chosen evaluation method must be capable of representing such constraints, either explicitly or implicitly, with the possibility that the constraint may be exercised constantly or only intermittently, that is it can be dynamically adjusted. Therefore, this element of the problem must be addressed when considering the choice of an evaluation method itself. The consideration of constraints is part of the normal behaviour of the evaluation. In contrast, should Level 3 be applicable then it must be accounted for directly in the evaluation definition stage as it will either result in the defined evaluation accommodating the constraint statically, e.g. leading to the definition of a reduced number of attributes, or it will require the definition of an adaptive form of evaluation. The final case will require a composite of the previous two treatments.

It may be possible to describe all known alternatives in detail, in which case the criteria given in Table 5.2 can be completed in full, generating an entry per alternative (that is for items ii and iii,

Table 5.2). If this cannot be done, or is thought unnecessary, then the description must remain of a general nature. It is considered that it will be the norm for decision problems arising within the context of system operations to handle alternatives which will be different instances of the same type, e.g. process threads. However, the identical nature of objects may not be immediately apparent, for example if trying to decide between two hardware resources such as a CPU and a network link, but given involvement in the same evaluation process a common level of abstraction should be identifiable at some point. The specific criteria which characterise an individual instance of an alternative reflect points raised in the general object description (Table 5.2, item i). Here, their number is kept to a minimum for convenience (i.e. only listing two).

An evaluation is intended to assess the state of two or more alternative objects, with the results of this assessment then being used to condition how the alternatives are subsequently treated. The issues of concern, which forms the focus of the attributes' attention, should be provided in the problem definition. Detailed, low level descriptions of a specific attribute may be provided if the decision maker has that information to hand, but if not the description should endeavour to be unambiguous in nature. A distinction can be made when defining attributes between those which directly address the issue of concern, and those which provide an indirect insight, the latter being commonly identified as *proxy attributes* [Kee76]. The stance taken here is that this is a secondary issue and can be overlooked with no significant operational consequence, so long as the choice of attribute can be justified.

The question of which object an attribute is associated with is covered in part by the role of the *owner* criterion (Table 5.3, item iii). The intention in providing this option is to indicate which object has responsibility for generating the attribute. Such an object may not be an alternative type, even though the subject matter which the attribute addresses is a property of the alternative. This situation describes, for example, the case where an alternative generates data which is subsequently handled by a second process and the second process attaches an attribute to that data. The same information might be retrieved by the second process as would have been generated if the alternative/producer had undertaken responsibility for the attribute directly. However, in the former case it is to be inferred that the alternative has no knowledge of the attribute and, therefore, does not have to give consideration to its support directly, rather this is the responsibility of the secondary process. Each of the intended attributes warrants its own tabular entry and should be relevant to the issues raised in the problem definition.

Attribute criticality (Table 5.3, item iv) encapsulates both the notions of significance and flexibility. The *mandatory* option describes the case where assessment of the attribute is essential to the appraisal of the objective and the production of an usable result. The *partially-*

mandatory option describes the case where assessment of the attribute is essential in some form, but an alternative measurement/representation to the normal variant is allowed. The *optional* case is where the issue represented by the attribute, although relevant to the problem, is not deemed fundamental to its satisfactory evaluation, that is to say it 'enriches' the investigation but a working solution to the problem can be arrived at even when it is omitted. For any evaluation the attributes may be either a mix or an exclusive set of the above types and the nature of this mix will have an effect on issues of modifiability and the level of coverage permissible. The ability to manipulate attributes may be influenced by the conditions under which they are employed. For example, if several attributes are defined, each investigating a distinct issue, then their criticality will be dependent upon the significance of the issue they represent. However, if more than one attribute reflects a different aspect of the same issue it might be deemed more feasible to ignore the less important elements on occasion. The *state appraisal* criterion (Table 5.3, item v) is the placeholder for the mechanics of the actual measurement methodology, i.e. a functional or verbal description of how the attribute is to be assessed and a quantitative representation of its state produced, with an indication of units, data type etc. if known. Establishing the degree to which an attribute will tolerate being traded-off against another attribute (Table 5.3, item vi) is envisaged as something of an ad hoc process which follows the lines of the decision maker considering under what conditions, if at all, he would prefer one attribute over another. It may be thought that the assumed preferences are related to the attributes attaining certain threshold values, in which case this should be noted. This is intended as an informal approximation to the full comparative appraisal of attributes which may be necessary subsequently and, as such, it raises an awareness of the potential existence of this condition in order that it can be accounted for when it comes to choosing a particular evaluation method. If attributes are tolerant of trade-offs then this increases the expressiveness of the evaluation method, likewise if they are not, then the final chosen method should not be such that it makes an assumption to the contrary. Again, it may only be possible at this point in time to approximately establish whether the worth of an attribute is value dependent or not (Table 5.3, item vii) but this is sufficient to raise awareness of the issue.

5.1.4 Forms of Modification

The overall cost involved in performing a multi-criteria evaluation of several objects will be a composite of several factors, including the size of the problem and the complexity of the individual attribute measurements. Whilst some of these limitations are unavoidable, it is assumed that others may be modified such that a reduced evaluation may still proceed under constrained circumstances. Three particular forms of modification are identified by which any inherent flexibility in the evaluation might be exploited, as given in Table 5.4. These are not seen as an exhaustive representation but rather as sufficiently broad concepts to support a general discussion.

i.	Truncation	Reduce the number of attributes so that only a partial set of those defined is considered in a particular evaluation instance.
ii.	Substitution	Temporarily, for the duration of one instance, substitute an element which is problematic with respect to the prevailing conditions with a less problematic alternative. Both attribute types and attribute measures can support substitution. Substituting an evaluation objective describes the act of switching objectives, a decision which is effected outside of an objective itself.
iii.	Skipping	Missing out an instance of evaluation [with/without application of a default behaviour].

Table 5.4: Dimensions of Flexibility

Truncation employs the notion of preferentiality, discussed elsewhere, as the basis for temporarily eliminating attributes from an evaluation instance. Modification, of whatever form, cannot be undertaken in an arbitrary fashion, however, as this will introduce inconsistencies in the treatment received by individual objects and also in the interpretations which can be applied to the results of different evaluation instances. Adjusting the number of objects being considered is excluded as an option as there is no means to do this other than on a random basis. The purpose of evaluation is to provide the data to perform this process of elimination in the first instance. The ability to drop an attribute is dependent upon its criticality.

Substitution of elements assumes that for some of the evaluation components there is a ‘cheaper’ alternative which may be temporarily used instead. For example, instead of employing a complex measurement of each attribute it may be acceptable to employ raw data on occasion. Substitution can be employed on attributes, attribute appraisal methods and evaluation methods but is done at a cost of a reduction in coverage and in the ‘quality’ of results. It may also require alternative methods to be loaded which, again, are not without cost. Substituting an evaluation objective should not be effected from within an objective as it is dependent upon external conditions which should have been assessed prior to commencing evaluation otherwise a redundant partial execution is generated which is inefficient.

5.1.5 Evaluation Rate

The selection problems considered in this discussion are assumed to be predominantly process-led operations, that is they are instigated by events and actions above the level of the attributes upon which their evaluation is based. They may feature multiple instances of multiple types of attributes, assessing a varying number of objects and with the possibility of supporting multiple objectives. They may also have a cyclical obligation to produce a result, given the repeated occurrence of a problem.

The central difficulty associated with attempting to assess the state of an object is that the act of observing and recording it is not normally associated with the ability to freeze the observed value at that point in time. Ensuring atomicity of actions, two definitions of which are provided below [Bur01], might be seen as one means of overcoming this problem.

> *"Actions are atomic if they can be considered, so far as other processes are concerned, to be indivisible and instantaneous, such that the effects on the system are as if they do not reveal their state changes until the action is complete."*

> *"An action is atomic if the processes performing it are not aware of the existence of any other active process, and no other active process is aware of the activity of the processes during the time the processes are performing the action."*

However atomicity may be both unrealistic and, also, incapable of providing a solution. In a multi-tasking system resources are shared and the activities of other processes, whilst hidden from each other, will have a collective impact. In the general case, whether operations are sequential or parallel, unless all processes involved in an evaluation activity have, from the commencement of measurement of the first attribute to the commencement of the activity which employs the results of an evaluation, exclusive access to the resources, then the underlying attributes may be subject to change as a consequence of the activities of other non-involved processes. This is particularly the case when some, or all, of the components of evaluation are not internal to the same process. If the production and consumption of values is contained within the same process then the disruption from other system activities may be less (depending upon the context) than if evaluation and consumption are split across separate processes.

Within a set of attributes the level of dynamism displayed may vary between the two extremes of being completely static to that of exhibiting continual change. In addition, the rate of change may be consistent across a single attribute type or different within that type. At runtime, all the alternative objects which are being evaluated may display highly fluctuating values or, indeed, only one of them may. This is a level of dynamism which is too complex to realistically be able to respond to in a system context, either efficiently or effectively, specifically when thinking in terms of trying to provide a numerical interpretation which captures the multiplicity of attribute states. It should be accepted, therefore, that some discrepancy between the current evaluation results and attribute states might be the norm unless a solution reflecting atomicity is found.

In the context of multi-criteria evaluation there are four particular scenarios of ‘changeability’ which may be encountered, as seen from the perspective of a newly instantiated evaluation process. These relate to the set of objects and the dynamic nature of their attributes when compared to any other evaluation instance:

1. Objects are changeable – attributes are static
2. Objects are changeable – attributes are dynamic
3. Objects are not changeable – attributes are static
4. Objects are not changeable – attributes are dynamic

Cases (1) and (3) do not present a problem as values, so far as they relate to an available alternative object, are static. The degree of fluctuation in objects may or may not be of concern, depending upon how the results are to be employed. For cases (2) and (4), however, there exists the potential for a discrepancy between the current evaluation result which forms the input to another process, and the real state of the attributes, and hence the alternative objects, which that process is concerned with.

To reiterate the description of an evaluation scenario; an evaluation result is produced which reflects the relative attractiveness of an object, with respect to a particular defined problem, by an analysis of the state of several attributes. One question of interest is whether it is possible to bound the minimum time for which a newly generated result will be in agreement with the underlying attribute values, before a change occurs in some portion of the latter. This can be interpreted as establishing the *worst-case* for the attribute's change rate, *aCR,* alternatively viewed as the longest period of coincidence between the inferred state associated with an evaluation's result and the actual real state of attributes. Taking the case of sequential execution where the evaluation process is periodic [sporadic] in nature with a periodicity [minimum inter-arrival time], T, and a computation time, C, and a set of attributes, each of which has an individual rate of change, aT_i, which might also be characterised as periodic or sporadic, where i is the ith attribute and where change is asynchronous across the set of attributes. A superficial assumption may be that if, for all i, $aT_i > T$, then the value of *aCR* will be higher, i.e. better, than if $aT_i < T$. In fact, for this scenario, the frequency of attributes has no significance to *aCR.* This is because, it cannot be guaranteed that at some point in the execution of the evaluation,

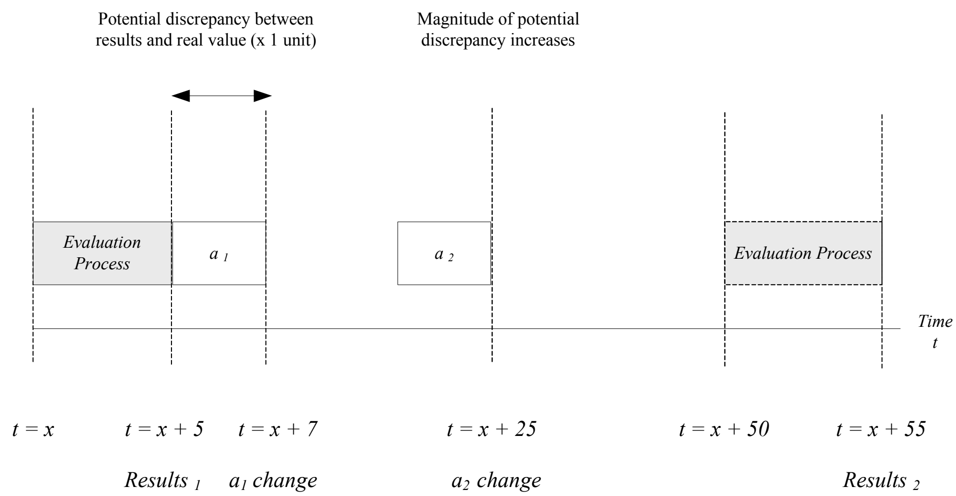


Figure 5.1: Discrepancy Between Evaluation Results and the Real Underlying State of Attributes

one of the attributes does not change its state immediately following the production of the latest evaluation result (Figure 5.1). Therefore, in the general case *aCR* will be equivalent to the computational unit/clock granularity of the hosting hardware. Attribute periodicity does influence, however, the number of instances which an evaluation can run without expecting that specific attribute to change.

On top of being expected to handle the inherent dynamism of attributes and their related issues, it should also be realised that even a change in a single attribute's value may be sufficient to bring about an alteration in the overall set of results. In addition, it may effect not only the object with which the changed attribute is associated but it may also impact upon the relative ordering of other objects, even where they, themselves, have remained unaltered.

Alternatively, attributes can change at any rate and be of no consequence to the evaluator if the new values are insufficiently large enough, or all of the same proportionality, that it brings about no alteration in the relative positioning of results.

The issues relevant to establishing whether a change in an attribute's state would be sufficient to bring about a change in the overall value, if an evaluation were to be run at that point in response to such a change, include:

- The actual set of attribute measurements which pertain at this point.
- The granularity of measurement which the evaluation method addresses.

- The number of attributes which have also changed since the last results were published, up to and including this point in time.
- The number of objects where change has occurred.
- The preferential significance of the changed attribute(s).

Again, accounting for the interaction of these factors and the combinations of changed attributes, makes it highly unlikely that this degree of changeability could ever be handled efficiently. Therefore, the approach taken here is that, when describing potential dynamic multi-attribute decision problems, it must be assumed that, unless clearly stated otherwise, some discrepancy between the real state of the system and that represented by the latest evaluation result, will exist. In light of this fundamental assumption it is proposed that practitioners defining these types of problems, are faced with the following options by which to overcome this problem:

I. Operate at a granularity which will experience discrepancies between value and state, but choose to ignore them, however large they might be, and allow the consumer to proceed regardless. This might be open to interpretation as 'gambling' on the problem's environment not changing during the inter-evaluation period. The ability to tolerate such an approach would be highly application specific.

II. Define attributes such that they have known, and supportable, rates of change. Such is likely to be the condition of the majority of real-world dynamic evaluation scenarios where the bulk of defined attributes describe the meta-characteristics of alternative objects.

III. Be pessimistic when undertaking attribute definitions and the identification of constraining values. That is, build into bounding values a margin of error/tolerance, with which to cushion subsequent behaviours from the negative effects of change. For example, if employing a global evaluation set particular stringent comparators.

IV. Use statistical definitions for attribute state appraisal such that evaluation becomes inherently probabilistic and uncertainty becomes an identifiable entity within the evaluation function.

Following on from the above proposals, it can be seen that trying to handle unfettered dynamism in an evaluation problem is infeasible. Even though one of the advantages claimed for enabling dynamic evaluation is that it will increase flexibility over static approaches, in

i.	Coverage Density	This provides an indication of how much of the defined normal evaluation is employed. See section [5.5]. Establishing the appropriate level of coverage is an exercise for decision makers and system experts. The potential usage is to act as a flag to increase importance of evaluation process if required.
ii.	Drift	This provides an indication of the distance from the ideal value coverage and update to that which currently prevails. See section [5.6].

Table 5.5: Criteria to Measure Impact of Modification on Evaluation

reality such flexibility must find itself compromised either in the definition or in the execution of the problem. Pessimism, as listed in item III above, has long been a known restriction on operations in real-time systems, whilst statistically driven behaviours may not be immediately compatible with predictability. In short, for most classes of problem scenarios where criticality is not so great that dedicated resources can afford to be accommodated, one or more of the above compromises must be accepted.

When discussing the implications of dynamism and the limitations identified in being able to deal with it, it may be of value to consider in the abstract what the perceived impact upon results would be arising from the changeability of certain attributes. To facilitate this discussion, a straightforward *pseudo-probability* mechanism is described in Section 5.3 with the objective of providing a framework around which subsequent investigations might be structured.

5.1.6 Evaluation Methods

In this discussion 'evaluation methods' is intended as a generic term to describe the various combinatorial techniques by which individual attribute measurements can be combined to produce an overall representation of an object's state with respect to the evaluation objective. Suitable elements might, therefore, range from simple additive operations to complex cost/benefit functions. Some examples of such techniques are provided in Chapter 4.

5.1.7 Impact of Modification

In establishing the cost involved in supporting an evaluation process, in addition to the normal concerns, e.g. runtime overheads and increasing complexity, there is also the issue of how alterations to the evaluation may impact upon the quality of the results produced. Two criteria are defined to explore this theme, listed in Table 5.5, and are discussed further in Sections 5.5 and 5.6.

5.1.8 Preferential Ordering

Some form of preferential ordering of attributes is common in multi-criteria evaluation to reflect differences in the perceived attractiveness of the issues addressed and the contribution each makes towards the form of a satisfactory solution. Whilst an explicit representation of such relative desirability may not be an inherent requirement of the evaluation method finally chosen, some form of relative assessment of the attributes is likely, even if only at an informal level. Preferences are explicitly required under this scheme, however, in order to guide dynamic modification by attribute reduction. An approach for the speedy identification of a preferential ordering is presented in Section 5.4.

An important factor when trying to establish the relative significance and role of objects involved in an evaluation process is trying to identify the underlying preference structure and set of, potentially subjective, beliefs which are associated with the evaluation problem. The more complex and larger the problem and the more objects involved, then the more difficult this can become. This issue needs to be considered as, under some approaches, it may be a major consumer of man hours/time in trying to define the method, e.g. the act of setting weighting constants. Potential questions relevant to this topic deal with issues such as the level of expertise available, how are interrogations of decision makers to be conducted consistently and is the problem actually worth the effort required in order to explore the subtleties of numerous alternative outcomes? It should be remembered that, when dealing with subjective issues, it makes little sense to think in terms of achieving an absolutely ‘correct’ answer. Therefore, without carefully structuring of this part of an evaluation process, if there are several decision makers providing input, it could extend well beyond the time allotted to it as disagreements emerge and opinions change.

5.2 Methodology

In order to approach the high-level analysis of a potential dynamic evaluation scenario, the problem has been decomposed into three sections. Section One deals with the provision of definitions, establishment of context, the clarification of terms etc.. Section Two encapsulates the analysis of the adaptive capability of the evaluation. Section Three undertakes the checking of the evaluation objective's requirements against the properties of possible techniques by which to undertake the actual evaluation activity itself. Figure 5.2 provides a diagrammatic representation of Sections One and Two.

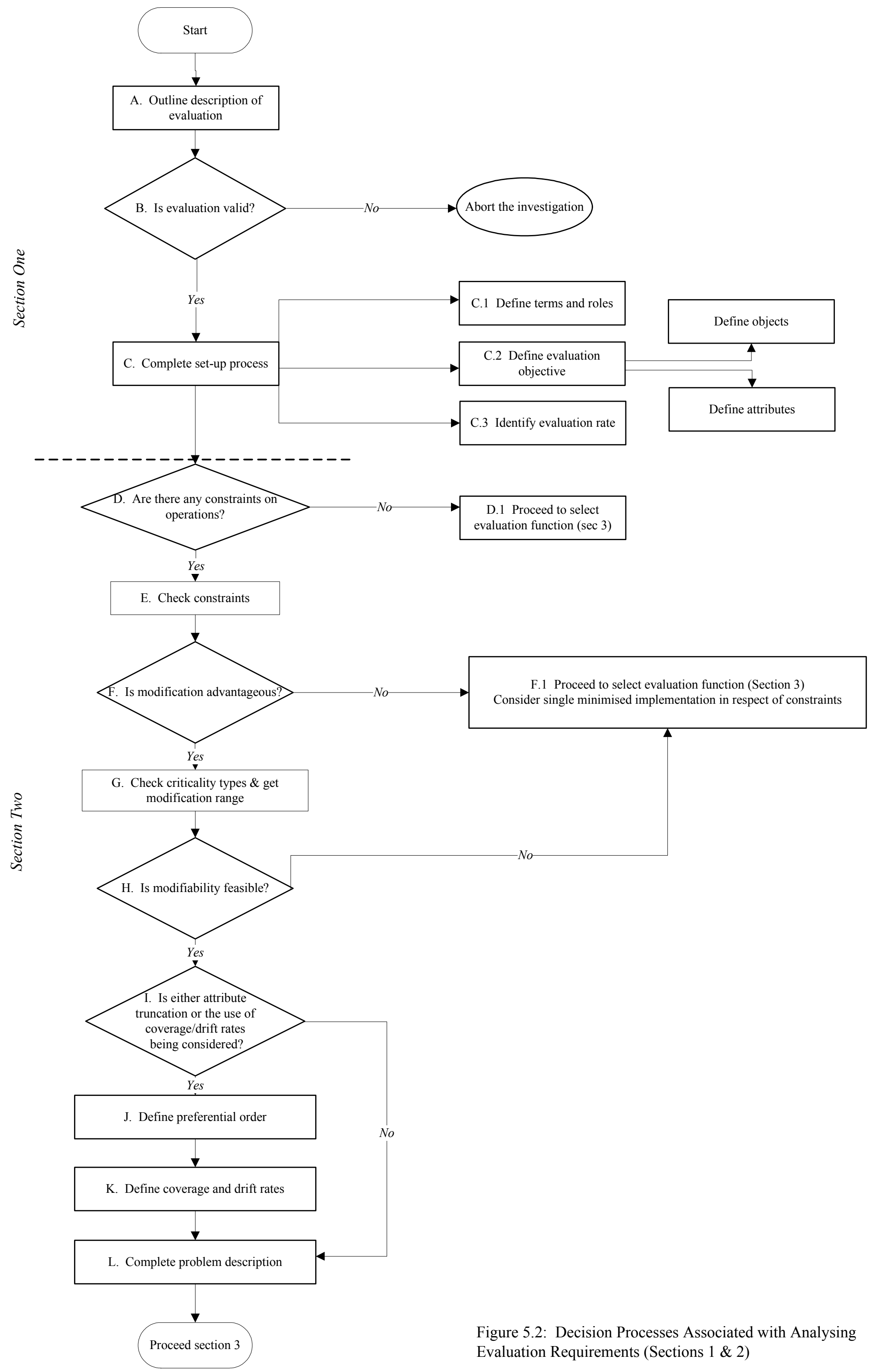


Figure 5.2: Decision Processes Associated with Analysing Evaluation Requirements (Sections 1 & 2)

5.2.1 Section One - Defining the Problem

The minimum inputs to this section are a selection problem, placed within a system context, together with extant knowledge of the components and context. This entails a description, with necessary detail, being provided of the perceived dynamic evaluation problem.

Perform validity check: As discussed earlier [5.1.2], before proceeding any further the question should be asked:

Q1.	Is Dynamic Evaluation Necessary?	
i.	Are the performance issues of concern all static and assessable prior to runtime? Y/N	If Y to i. and ii. then dynamic evaluation is not required. For all other cases, dynamic evaluation is required.
ii.	Are all instances of alternatives known and assessable prior to runtime? Y/N	

Complete set-up process: Provide the definitions of objectives and entities, together with customized terms, and rate of change [5.1.1, 5.1.3, 5.1.5]. Any operational constraints which are to be accounted for within the evaluation need to be flagged, even if the precise value is not known at this stage. Constraints can include 'real' parameters relating to functional requirements of the consuming process. They might also encapsulate potential general problems of the environment, such as high levels of utilisation or node failure. After completing the definitions phase it may be beneficial to consider Question (2) in order to obtain an overview of possible conflicting behaviours and interactions between the attributes.

Q2.	Do Defined Attributes Tolerate Trade-off?	
i.	Are there multiple attributes? Y/N	If N, then stop, as single attribute and therefore trade-off internal to evaluation not an issue.
ii.	Are all attributes trade-off tolerant? Y/N	If N, need to consider whether it is worth partitioning the evaluation in separate sectors of like-attributes to facilitate different treatments or re-defining attributes to remove conflict.
iii.	Are there constraints on trade-off tolerance? Y/N	Need to consider the existence and representation of threshold values associated with attributes.

On completion of Section One, the evaluation problem will have been defined as will the characteristics of the objects and attributes by which it is going to be resolved. An indication of the changeability and constraints which the evaluation will be subject to should also have been gained.

5.2.2 Section Two - Considering Adaptive Capabilities

The minimum inputs to this section are the completed definitions of components and identified constraints. Without having a thorough knowledge of the properties of the elements involved in the evaluation, particularly the properties of the suite of attributes, then it is difficult to continue to tackle the issue of possible adaptability. The advantage of considering the twin strands of dynamic evaluation and adaptability concurrently is that it makes for a more integrated behaviour. If attempts at adaptability were to be superimposed onto an already extant evaluation then the changes may not necessarily be readily absorbed.

Check Constraints: As the discussion is taking place prior to implementation, then it is assumed that there is only one factor upon which to base the decision to make an evaluation modifiable, namely whether possible impediments to performance have been foreseen as a consequence of the activities carried out in Section One. Hence, a check on the existence of a constraint defined under Section One is sufficient to determine whether an investigation of modification issues should proceed or not. Constraints reflect bounds upon operations and stem from both internal and external conditions. Therefore, if the restrictions they impose can be ameliorated by adaptive techniques then it may be possible to extend the operational niche which an evaluation process can occupy. Alternatively, it may be that the constraints are to be permanently accounted for in the static definition of the evaluation. Following from this if, on reconsideration, a constraint is deemed to constitute an insufficient impedance to normal operations, then it should be treated as of no concern with respect to this specific issue. Therefore, the options available are:

Constraint applies - yes → *modification advantageous*
Constraint applies - no → *modification not required*

Check Feasibility and Type of Modification: Establishing that enabling an evaluation to exhibit an adaptive response is desirable is of little advantage if the nature of the problem is such that such behaviour cannot be accommodated without undermining the validity of the evaluation. This gives rise to Question (3):

Q3.	Will the Evaluation Support Modification?	
i.	Are all attributes mandatory? Y/N	If Y, then stop as modification not tolerated.
ii.	Is evaluation critical? Y/N	If Y, then cannot employ skipping.
iii.	Is there more than one attribute? Y/N	If Y then may be more likely to be able to exploit dropping of attributes, dependent on its type.

This query can be resolved by considering the interaction of objects' and attributes' properties. The restrictiveness associated with different characterisations enables certain forms of modification to be favoured and precludes others, as shown in Table 5.6.

Level of Criticality	*Form of Modification*			
	Number of Attributes	*Evaluation Frequency*	*Substitution (Type)*	*Skipping*
Critical Objective	n/a	N	n/a (1)	N
Non-Critical Objective	n/a	Y	n/a	Y
Mandatory Attribute	N	n/a	N	n/a(2)
Partially-Mandatory Attribute	N	n/a	Y (measurement)	n/a
Optional Attribute	Y	n/a	Y	n/a

Notes:

(1) Substitution of an objective is equivalent to running an alternative objective

(2) For attributes, the notion of skipping an attribute is equivalent to dropping it under the process of reducing attribute numbers.

Table 5.6: Suitability of Particular Dimension of Modification to Levels of Criticality

Evaluation Objective Criticality	*Criticality of Set of Attributes*						
	All Mandatory	*All Partially-Mandatory*	*All Optional*	*Mixed Mandatory/ Optional*	*Mixed Mandatory/ Partial*	*Mixed Optional/ Partial*	*Mixed Mandatory/ Partial/ Optional*
Critical	none	3b	*	1, 3ab	3b	1, 3ab	1, 3ab
Non-Critical	none	3b	1, 2, 3ab	1, 3ab	3b	1, 3ab	1, 3ab

Key: 1 – Adjust number of attributes; 2 – Skip evaluation; 3a – Substitute different type of attribute; 3b – Substitute different form of measurement for current attribute type.

Notes: * This variation is inconsistent with the definitions: if an evaluation is critical it should not be characterised entirely by optional attributes.

Table 5.7: Allowable Forms of Adaptation under Criticality Constraints

Using the contents of Table 5.6, it is then possible to identify the appropriate types of adaptive response to use under any particular combination of attribute and objective criticality. Table 5.7 shows those types of adaptive response which are allowed in each situation, given the stated properties of the attribute set. To re-iterate, attribute criticality is defined with respect to the attribute's significance in assessing the evaluation objective. So, for example, if an evaluation is non-critical and all attributes are classified as partially-mandatory then modification is restricted to the substitution of the type of attribute measurement. The choice of adaptive behaviour to pursue will be dependent upon a particular context however it is apparent that allowing the problem definition to be quite relaxed about the critical and mandatory nature of its components provides for the greatest flexibility in this respect. Owing to the dominant nature of the mandatory definition, skipping is only feasible in one of the illustrated combinations, namely where the objective is non-critical and all the attributes are optional. Deciding whether to follow a course of employing either attribute truncation or substitution, or indeed skip an evaluation instance completely, will be dependent upon the considerations of a particular case. However, considering the level of coverage required will help reveal which actions are permissible and which not.

If the intention is to consider the use of either attribute truncation or the specification of coverage rates, then it is necessary to next consider some form of preferential ordering by which to distinguish the defined attributes. If neither of these two options is to be considered or if modification is to be applied in an arbitrary manner, then the analysis can proceed directly to Section Three.

Define Preferential Order: As has been stated, preferences have a role in resolving multi-attribute trade-off problems and also in representing subjectivity at attribute level. Here, they are being employed for a third reason, namely to identify which attributes to drop if the number being evaluated is to be temporarily reduced. In the case of only a few attributes, the preferential relations between them may be sufficiently well-known to permit the ordering to be arrived at easily. In more complex, or unfamiliar, problems identifying a coherent set of relations may not be such a straightforward task. Section 5.4 describes an approach designed to support the latter situation.

Define Coverage Rates & Drift Limits: This notion was briefly described in section 5.1.7. One interpretation as to how this might be carried out is discussed in Sections 5.5 and 5.6. The purpose in considering the amount of coverage required of an evaluation process is to allow the on-going nature of the evaluation to be exploited. With a static, one-off, evaluation there is no opportunity to consider the time-variant levels in the quality of an evaluation which might be tolerated. Coverage is defined to explore this aspect and uses the number of attributes and frequency of evaluation as indicators of overall evaluation ‘quality’.

Complete the Problem Description: The concluding operation of Section Two is to collate the information produced so far which describes the decision problem, highlighting the salient features, in order that these can be compared against the properties of possible evaluation methodologies.

5.2.3 Section Three - Selecting Appropriate Method

The third stage of the analysis involves the matching of the known details of the evaluation, as gleaned from the activities of Sections One and Two, with the known properties of multi-attribute evaluation techniques. The decision maker will either know of the individual capabilities of the range evaluation techniques being considered, or will have to investigate the matter prior to this point, for example as is done in Chapter 4. In summary, the inputs into this section are as follows:

- A set of definitions by which to characterise the identified evaluation problem, together with an indication of the properties it exhibits and an identification of the various elements which are participants in the evaluation process.

- An indication of the degree of flexibility of the evaluation, the forms of adaptation which are most appropriate to enable and a set of tools by which to manage the adaptation process.

- A consideration as to possible constraints upon the undertaking of evaluation, either with respect to the environment or the evaluation itself.

- A selection of two or more potential candidate methods by which to implement the evaluation.

The material collated so far in the course of analysing the identified problem will be of benefit either to the task of choosing an appropriate evaluation method or become relevant during the subsequent implementation phase. Therefore, it may not all be directly employed during this stage. Section Three involves two separate activities, that of deciding upon the group of alternative evaluation techniques to consider further and on actually undertaking the comparison between the problem requirements and the evaluation methods' capabilities.

Identifying the Set of Alternative Evaluation Methods: As has been said, the source of information on evaluation methods can either be extracted from common practice or derived from analysis along the lines of that undertaken in Chapter 4. The first step in this section is to consider which of the many techniques that are capable of dealing with a multi-attribute decision problem are to be expressly considered further and to ensure that a sound understanding of their manner of operation, expressiveness and limitations is in evidence. An argument might be made that formal analysis intent on characterising evaluation methods should form part of the definition activities performed in Section One. However, by having carried out an interrogation of the decision problem and the evaluation requirements, the decision maker has a greater awareness of the issues involved, together with the nature of the problem and the type of questions which need to be asked of candidates for the role of evaluation methodology, thus making for a better targeted selection of the more suitable methods if it is done at this point.

The term ‘evaluation method’ is used in a general way to represent any process which is deemed capable of undertaking the appraisal of a set of individual attributes and presents a result in a form which allows the attributes' collective state to be used to interpret the state of the parent object with which they are associated. This definition is general enough to allow for a range of possibilities to be considered, from straightforward type-checking operations to complex constraint satisfaction algorithms. In Chapter 4, three forms were looked at in particular, the MCDA weighted additive value method, an approach based on enumeration and scoring of the significant criteria states and a method which favours the lexicographic treatment of criteria.

The ability to come up with several different techniques which might be suitable for further consideration will be dependent upon the knowledge and experience of the participants in the design process. There are two ways in which the problem of the identification of evaluation methodology can be approached. It could be that only a single method is presented at first, either because it is familiar and deemed sufficient for the task or because no other methods are known of. In this case, a direct comparison with the properties and requirements of the evaluation objective should reveal where the two elements are in agreement and where features of the evaluation cannot be supported by the proposed method. As a consequence of this, the decision maker is left with the option of sticking with his single choice of evaluation method but accepting that it will restrict the expressiveness and/or flexibility of the evaluation objective when finally realised. Alternatively, he can seek a second possible technique and repeat the comparison process to see if the consequences improve. The other way of approaching the specification of alternative techniques is to make available at the start of the comparison stage a pool of possible candidates and to methodically proceed through these in full, making the final choice of evaluation methodology on the basis of the best perceived pairing between problem and method. Although more time consuming, expertise and familiarity with the concepts involved would, it is felt reduce the complexity and allow for better coverage of the comparison process.

Naming the set of alternative methods to be considered involves an informal, high-level review of the properties of the proposed methods, primarily considering the fundamental structure of their operation and the form of their output, in order that these details can then be superficially compared with the type of selection required, and the less suitable techniques dropped.

Comparison of Evaluation Requirements against Evaluation Methods: The second part of Section Three requires a more detailed investigation of the suitability of the possible evaluation methods. There are three areas of particular interest and which represent the bulk of the concerns in selecting an appropriate method. Firstly, the ability of a methodology to capture and express the properties of the evaluation objective, specifically with respect to issues of attribute trade-off and expressions of preference as applied to the defined attributes. These would be common requirements of any multi-attribute evaluation technique. Additionally, however, there is the question of dynamism to be considered and whether techniques which are amenable to the static exploration of such decision problems, are equally suited to representation in a dynamic form. This point is related to the third area of concern, that is any technique's ability to cope with the range of adaptive behaviours which have been described. In this respect, the ability to support the ‘skipping’ of an evaluation can be overlooked as any tolerance in this area is a property of the consuming process, not the evaluation method per se (each evaluation instance is complete within itself and therefore omission has no bearing upon the integrity of a single set of results).

It is assumed that the process of comparison will take the form of applying a series of questions in natural language by which to interrogate the candidate methodology. The focus of questioning is supposed to be upon the sufficiency and flexibility of any intended evaluation method but it may be opportune to raise general operational questions at the same time. Questions should address the issues which have been identified so far, for example:

1. How appropriate is this evaluation method to performing the type of selection required (ranking, reduction, categorisation)?

2. How amenable is this form of evaluation method to the possible types of modification?
3. What would be the mechanical effects of modification on the usability, the integrity and the validity of the results produced if the named form of modification were to be implemented directly with no extension to the method's definition?
4. How much work would be involved in investigating, specifying and implementing modifications and extensions to the method's definition?
5. What are the additional runtime overheads likely to be for the modified version over a non-modified version of a particular evaluation method?
6. How dynamically scalable is the function and what are the likely overheads to be given the known number of attributes and alternative objects stated in the objective definition?
7. How inherently robust is the evaluation method, that is to say, what are the levels of integrity and usability associated with a set of partial results, as might be generated if the process is interrupted?

The above questions are not definitive but are representative of the types of queries which must be used to investigate the three areas of concern as stated previously. Dealing with each of the possible techniques in turn, a picture of their relative suitability can be ascertained. Answers may be represented in natural language or make reference to a pre-defined verbal or numerical scale, e.g. good, medium, poor, with items and ranges relevant to each question, which itself might be custom-defined for this evaluation problem scenario or an industry standard form used in all similar situations. The depth of interrogation would be particular to the problem and would, in part, be dependent upon the previous experience of the decision maker with problems of this type. That is, if his expertise is such that he can readily grasp the issues involved, then he might bypass some of the formal and pedantic questioning process. However, there is an advantage in endeavouring to make the comparison as precise as knowledge will allow as the more specific are the descriptions and responses given then the more information is available by which to distinguish between alternative methods. Even in the case where the capabilities of the functions are deemed approximately equivalent, the process of analysing the evaluation may be of benefit in helping decompose the problem and highlighting any inconsistencies present.

When implementing an evaluation problem, there are two potential sources of problems, one arising from the efficacy and integrity of the evaluation problem as defined, and the other from operational difficulties such as location of objects and attributes.

By the conclusion of this section, the inherent nature of the evaluation problem should be clear and a choice made as to the most appropriate method for the dynamic treatment of it, including the implementation of any adaptive behaviours, if feasible. The bounds on adaptability, if any, can be described by the specification of coverage and drift values. In addition, there will be a set of definitions and data by which to guide the subsequent development task.

5.3 Pseudo-Probability of Change

As a precursor to simulation activities, one can consider the level of disruption inflicted upon an existing set of results if one, or more, of the defined attributes were to exhibit a change in its state such that, if it were to be reassessed immediately following the change having occurred it would generate a different value which in turn would give rise to a re-ordering of alternatives. Taken at an abstract level this problem can be simplified by redefining it so that it makes use of a preferential ordering, if any is present, rather than observed probability distributions. For example, for a single attribute changing and, given its stated significance to the satisfactory appraisal of the evaluation problem (inferred from its preferential position), then what would be the probability of its significance being sufficient, in the general case, to bring about a change in the final result if the evaluation were to be re-run immediately after the attribute had changed? The scheme is based on the premise that, regardless of the chosen evaluation methodology, given a preferential ordering of the attributes to be assessed then it is reasonable to suppose that the greater the level of preference for an attribute then the greater will be that attribute's influence in the evaluation. The proposed scheme assumes that if several changes occur at the same instance then only the most highly preferred attribute need be considered as it will have a dominating effect. If it is thought to be of insufficient interest then neither will the lower ordered attributes. This reduces the number of combinations of attributes and objects which need to be accounted for down to the number of attributes, w, multiplied by the number of objects, n. If, when no attributes change on any object the probability of a value change is 0, then when all attributes change on all objects, the probability of a value change is 1. The attributes are ordered on a preferential basis, in this case $a_5\ P\ a_4\ P\ a_3\ P\ a_2\ P\ a_1$. If the attributes did not accommodate a preferential ordering then the number of attributes changed might be substituted for the ordered IDs. Given a pairing of <attribute change, number of objects registering a change>, then for $w = 5$ and $n = 5$ there are 25 pairs to be considered, representing 25 different states. Each represents an incremental worsening of the impact their change will have and the pairs can be ordered from least significant, identified as $< a_1, 1 >$, next least significant as $< a_1, 2 >$ and so on until the maximum impact is achieved with $< a_5 , 5 >$. A notional interpretation of the probability that a change in the stated attribute at the stated level will cause a change in the overall results can then be provided by taking the *pairing's ordinal position/total number of pairs*, e.g. for a change in a_1 which affects only one object then a probability of 1/25 is given, meaning it has a minor probability of generating disruption when compared to a_4 changing its value on 5 objects which would generate a pseudo-probability of 20/25. Table 5.8 illustrates the pseudo-probabilities for this case.

No change - P = 0	*Number of Objects Changed*				
	n = 1	*n = 2*	*n = 3*	*n = 4*	*n = 5*
a_1	.04	.08	.12	.16	.2
a_2	.24	.28	.32	.36	.4
a_3	.44	.48	.52	.56	.6
a_4	.64	.68	.72	.76	.8
a_5	.84	.88	.92	.96	1

Table 5.8: Pseudo-Probabilities Associated with Change of Attributes and Number of Affected Objects

This mechanism does not, of course, provide the real probability of an attribute causing a re-ordering of an existing result but it does provide a tool by which to discuss the issue of change impact in the abstract, independent of the complexity of managing any real values and with respect to the inherent significance of an attribute. As has already been noted, even a single instance of an attribute changing can give rise to a change in the order of results.

One possible usage of such a scheme is that it allows the evaluator to express what theoretical level of activity he is interested in by setting a probability of disruption threshold to a level appropriate for the importance of the activity being managed. Such information can be fed through to the implementation stage to inform activities there. Above this threshold the disruption in an attributes' state is thought to be of sufficient interest to warrant undertaking a re-evaluation of the alternatives. For example, if an arbitrary threshold level of 0.5 were set then all pairings from $< a_3, 3 >$ onwards would instigate a re-evaluation. However, whether at runtime a re-evaluation is worth undertaking will be dependent upon the context in which evaluation is employed. If the consuming process is highly repetitive then the level of change might be used to reduce the evaluation burden by setting its frequency to approximately reflect that anticipated of the disruption threshold, or tying it specifically to be triggered by the same. Likewise, if the results are used as input to a task which is open-ended then if re-evaluation is triggered it might be effected straight away. In contrast, if the results are used to guide a task which, once started runs to completion then there is no point in re-evaluating. For this reason, tying the disruption threshold to the least of the mandatory attributes would not necessarily be of any benefit.

5.4 Preferential Ordering Heuristic

If an evaluation is, of itself, going to be dynamically adaptable then any approach based upon modifying the number of attributes required will need some means to identify how this modification should be undertaken. Here, it is proposed to use an ordinal definition of preference as applied to the set of attributes, the intention being to employ it to structure one particular form of adaptive response within the evaluation. Although the approach may be of benefit when defining the chosen method of evaluation, given its intended usage it is assumed that it neither needs nor, indeed, can take account of, trade-offs between attributes on the basis of the values they hold. Trade-offs within an evaluation are made on the basis of knowledge about the relative values of attributes, which cannot be known prior to undertaking the evaluation, which in this framework occurs after the decision on the level of adaptability of an evaluation is made. Any perceived trade-offs between attributes arising as a consequence of changes in conditions external to the evaluation are reflected in the definition of different evaluation scenarios.

An ordering of attributes is necessary so as to provide for a systematic approach by which they can be identified for exclusion if modification by reducing the number of attributes being appraised is to be supported. Desirability can be interpreted as a preference for one attribute over another and any inherent ordering will emerge if the relations between all pairs of defined attributes are directly considered with respect to this issue. If inconsistencies emerge during the definition process, particularly with respect to the identification of the relative significance of each attribute, then resolving these by reiterating through procedures will cause extra work. If the evaluation being defined addresses a complex problem then this might be justified by the importance and difficulty of the project. In situations where the problem is more constrained, however, and plays only a minor role in a larger operational concern, then of necessity the emphasis may fall on the speedy, but adequate, resolution of such problems rather than allowing for a slow but thorough handling. If only a small number of attributes are defined, dealing with readily understood issues, then there is unlikely to be a problem. If the problem is more complex or the criteria represent obscure themes then preferences may be harder to capture. Therefore, in order to reduce the possibilities of inconsistencies being present in the results after all preferential relations have been evaluated and to facilitate the identification of problematic pairings, which would threaten the production of a consistent ordering, the following heuristic methodology for identifying a preferential ordering is suggested. It focuses on identifying the specific range of options available to the decision maker when he is considering the next undefined pairing of attributes, given the relations he has identified so far. It also aims to reduce the number of pairwise comparisons which it is necessary to explicitly consider by directly exploiting the known properties of the relations already defined.

Description of Process

The following notations are employed:

Set A $= \{ x \mid x \in$ all attributes defined for appraisal under this evaluation problem$\}$

The pool of relations available is:

$P , nP , I , J: \quad A^2$

P	$\triangleq$	$\{ < a_1 , a_2 > \mid$ for $a_1 , a_2 \in A, a_1$ is strictly preferred to $a_2 \}$
nP	$\triangleq$	P^{-1} , that is $\{ < a_1 , a_2 > \mid$ for $a_1 , a_2 \in A, a_2$ is strictly less preferred than $a_1 \}$
I	$\triangleq$	$\{ < a_1 , a_2 > \mid$ for $a_1 , a_2 \in A, a_1$ is indifferent to $a_2 \}$
J	$\triangleq$	$\{ < a_1 , a_2 > \mid$ for $a_1 , a_2 \in A, a_1$ is incomparable with $a_2 \}$

P, I and *J* are the core relations employed in preference models. The inverse of *P* is added to facilitate specification in the matrix structure employed here. The properties defined on each relation are:

P	antisymmetric, transitive, irreflexive
nP	antisymmetric, transitive, irreflexive
I	symmetric, transitive, reflexive
J	symmetric, irreflexive

For all $< a_1 , a_2 >$ pairing of attributes, one and only one, of the above relations must apply.

Preferential relations can only be established if considered specifically in the context of a meaningful problem and are not applicable outside of this problem. The semantics applicable to preferences as applied here are that they relate to all attributes independent of the values which they might hold. The definition of *I* as transitive prevents the consideration of relations between a pair of attributes taking into account threshold values below a certain value of which the relation is one of indifference and above which it is one of preference.

In order to structure the undertaking of the pairwise comparison of attributes a matrix with the elements of *A* acting as indices along both rows and columns is defined. Indices to rows represent the domain and indices to columns the co-domain of the relation respectively. Starting on the top row and ignoring the main diagonal and all spaces to the left of it, each pairing is considered in turn, the appropriate relation identified and added as that entry. When the top row has been completed then the evaluator moves to the next row down and to the first entry to the right of the main diagonal, as before. The process continues until all empty slots to the right of the main diagonal have been completed. From this completed half of the matrix, the remaining half can be inferred using the property of symmetry as defined on their mirrored twin. That is, if either,

$$a_1\ I\ a_2 \text{ or } a_1\ J\ a_2 \qquad \text{then} \qquad a_2\ I\ a_1 \text{ and } a_2\ J\ a_1$$

or,

$$a_1\ P\ a_2 \text{ or } a_1\ nP\ a_2 \qquad \text{then} \qquad a_2\ nP\ a_1 \text{ and } a_2\ P\ a_1$$

respectively. The main diagonal may be populated employing the property of reflexivity but is not strictly required. The rows and columns must employ the identical ordering of attributes though these can be arbitrarily listed in the first instance. The process should commence with the longest row which has the greatest number of undefined relations to the right of the main diagonal, that is on the top line.

In identifying the top-most row of relations, there are no restrictions upon the choices available to the evaluator. However, if consistency in opinion is to be maintained, then all subsequent rows must take into account the previously identified relations and the properties they exhibit. Taking the case of three attributes, $\{ a, b, c \}$ and the three relations between them, when two of the relations are known and the third remains to be established then the latter, when it is named, should be consistent with the belief structure of the other two. For example, in Figure 5.3, the unknown relation being queried currently, designated *R*, is shown with the arrows running from the domain to the co-domain of the relation. If the decision maker were to name *R* such that it became $b\ P\ c$, then this would indicate inconsistency in thought as it would establish a circuit of relations, $a\ P\ b\ P\ c\ P\ a$ which clearly is flawed in its logic.

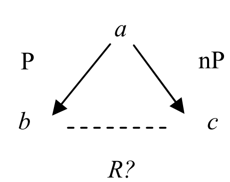


Figure 5.3: What is the Consistent Choice for Relation *R*?

That such a situation could genuinely arise might reasonably be expected and, if the decision maker were to persist in his opinion then the problem would require further investigation in order to clarify the underlying beliefs. This situation is allowable, however the problem that is being addressed here is one where the decision maker incorrectly cites a relation, possibly as a consequence of having temporarily overlooked his previous statements. Such an occurrence is seen as a genuine mistake and it is these which this method is aimed at speedily resolving. In order to provide guidance, without deliberately constraining a decision maker's options, Table

5.9 shows the sub-sets of permissible relations for the third unknown relation given the two relations already defined upon an identical domain. The operation can be compared to that of taking ‘sightings’ in yachting in order to establish the current position. Each of the higher rows will produce a pair of relevant relations which can be cross-referenced with Table 5.9 to identify the set of permissible relations with which that pairing is associated. The set of relevant relations consists of all those relations which map attributes whose domain is in a higher row to the current attribute together with all the relations which map these higher domains to the attribute representing the co-domain of the current query. Thus, as the analysis progresses down the rows in the matrix, the number of relevant relations will increase. So, to illustrate the case of a_3 R a_4, where R is as yet unknown, as shown in Table 5.10, there are four relevant relations, each identified by an asterisk, giving rise to two sub-sets of permissible relations as shown in Table 5.11.

Given the four defined relations, n = 4, there are 16 different pairings which need to be accounted for in Table 5.10:

$$\textit{number of different pairs of relations} = \frac{4!}{(4-2)!} + n = 16$$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1	X	P	nP*	I*	P	J	P
a_2	!P	X	nP*	nP*	P	P	I
a_3	P	P	X	**?**	-	-	-
a_4	I	P	-	X	-	-	
a_5	nP	nP	-	-	X	-	-
a_6	J	nP	-	-	-	X	-
a_7	nP	I	-	-	-	-	X

Table 5.10: Establishing the Relations Relative to a_3

Previous Rows	*Relevant R_1*	*Relevant R_2*	*Tuple of Relevant Relations*	*Permissible Relations*
Row 1	a_1 *nP* a_3	a_1 *I* a_4	< nP, I >	{ P, J }
Row 2	a_2 *nP* a_3	a_2 *nP* a_4	< nP, nP >	{ P, nP, I, J }

Table 5.11: Relations already Identified of Relevance to Establishing a_3 R a_4

Therefore, if the set whose elements consists of all the sub-sets of permissible relations identified so far as relevant to R is Q_r, and the specific sub-set of permissible relations which are available to R is $perm_r$ then,

	Relations Graph:	Defined Relations < a R b, a R c >	Permissible Relations { b R c }
1.	a b ---- c	< P, P >	{ P, nP, I, J }
2.	a b ---- c	< P, nP >	{ nP, J }
3.	a b ---- c	< P, J >	{ P, nP, I, J }
4.	a b ---- c	< nP, P >	{ P, J }
5.	a b ---- c	< J, P >	{ P, nP, I, J }
6.	a b ---- c	< P, I >	{ nP, J }
7.	a b ---- c	< I, P >	{ P, J }
8.	a b ---- c	< I, I >	{ I, J }
9.	a b ---- c	< I, nP >	{ P, J }
10.	a b ---- c	< I, J >	{ J }
11.	a b ---- c	< nP, I >	{ P, J }
12..	a b ---- c	< J, I >	{ J }
13.	a b ---- c	< nP, nP >	{ P, nP, I, J }
14.	a b ---- c	< nP, J >	{ P, nP, I, J }
15.	a b ---- c	< J, nP >	{ P, nP, I, J }
16.	a b ---- c	< J, J >	{ P, nP, I, J }

Key:

- $a \longrightarrow b$ *a* is preferred to *b*
- *a* *b* *a* is NOT preferred to *b*
- *a* *b* *a* is indifferent to *b*
- *a* *b* *a* and *b* are incomparable
- *a* ------ *b* the relation between *a* and *b* which is to be identified

Table 5.9: Defined and Permissible Relations for Three Elements

for $\forall\ x \in Q_r$

$$perm_r = x_i \cap x_{i+1} \ldots \cap x_n$$

where the value of n will be equivalent to the number of rows above that upon which R is located. So, for the example R shown in Table 5.10,

$$perm_r = \{P, J\} \cap \{P, nP, I, J\} = \{P, J\}$$

meaning that if the decision maker feels that R is either a relation of preference or indifference then his choice will be consistent with the relations expressed so far. To re-iterate, the decision maker is not expected to choose one of the permissible relations if it does not accurately represent his true opinion, so the mechanism should not restrict his expressiveness. They merely serve to facilitate the resolution of the problem involved. The maximum number of sub-sets of permissible relations which are required to be identified for any single domain criteria is equivalent to the number of relations to be established lying to the right of the diagonal multiplied by the number of rows lying above the domain's row.

To illustrate with a general example, consider seven criteria which are identified as being of significance in the selection of a holiday:

A	Price Reduction - this is a cost related attribute presented in this form in order to preserve uniformity in the evaluation with all other values i.e. larger value is preferred to smaller value for all attributes
B	Sunshine
C	Temperature
D	Food Quality
E	Sea
F	Historic Monuments
G	Sporting Activities

Starting with the top row, the pairings to the right of the diagonal are considered in turn. The top row can be considered without referring to Table 5.9, however all subsequent rows should follow the approach previously applied. An example for the case of *E R F* is shown in Table 5.12 and the completed example provided in Table 5.13.

Relevant R_1	*Relevant R_2*	*Defined R*	*permR*
A nP E	*A nP F*	$< nP, nP >$	$\{ P, nP, I, J \}$
B P E	*B P F*	$< P, P >$	$\{ P, nP, I, J \}$
C nP E	*C P F*	$< nP, P >$	$\{ P, J \}$
D nP E	*D nP F*	$< nP, nP >$	$\{ P, nP, I, J \}$
		$Q_r =$	$\{ P, J \}$

Table 5.12 : Identifying *E R F*

If on any occasion the set of permissible relations does not contain one that the decision maker feels is appropriate then there is the option of either discontinuing or reviewing all the defined relations up to the current point to see whether a change in opinion is required. The entries can then be altered and re-worked from the changed point forward.

	A	***B***	***C***	***D***	***E***	***F***	***G***
A	X	nP	nP	P	nP	nP	nP
B	P	X	P	P	P	P	P
C	P	nP	X	P	nP	P	nP
D	nP	nP	nP	X	nP	nP	nP
E	P	nP	P	P	X	P	P
F	P	nP	nP	P	nP	X	nP
G	P	nP	P	P	nP	P	X

Table 5.13: Identified Relations for the Holiday Evaluation Criteria
(P = preferred; nP = not preferred; X = not required)

	P	**nP**	**I**	**J**
A	1	5	0	0
B	6	0	0	0
C	3	3	0	0
D	0	6	0	0
E	5	1	0	0
F	2	4	0	0
G	4	2	0	0

Table 5.14: Count of Relations per Attribute

An approximation of the ordering can be quickly gained by counting the number of each relation type achieved by each attribute (Table 5.14) and then sorting on the basis of the values obtained for the counts against P, the bigger the value then the higher up the order will be that attribute. In the case of ties the state of the nP column can then be considered, on the basis of the lower the value the more preferred is the attribute, in order to differentiate between the equivalent attributes. The J and I columns need not generally be considered directly when deriving an approximation of the ordering.

The number of relations which must be established by direct consideration and for the situation where the main diagonal is not considered is, where n = number of attributes,

$$\textit{number of relations} = (n-1) + (n-2) + \ldots + (n-n)$$

which in the case of $n = 7$, means 21 comparisons are made. In order to reduce the number of comparisons required the assumption is made that the relations to the left of the diagonal are the inverse of those already specified to the right. The decision maker, whilst completing this portion of the table is relied upon to notice if the prompted relation does not tie in with how he actually feels about a particular pairing. The actual amount of work required, however, if this mechanism is employed to establish a preferential ordering is greater as each new relation to be positively identified will require two relative relations on each of the previous, higher order, rows to be looked up and the associated set of permissible relations retrieved. This, however, is a mechanical process and, therefore, can be completed more quickly than if time has to be spent pondering on all possible options. It also serves the dual purpose of coincidentally ensuring that the finished table of relations will be consistent, if consistency is a state which the decision maker's opinions support. Decision techniques requiring a full pairwise comparison, e.g. AHP, would generate n^2 comparison operations, but then may not offer any consistency checks and, consequently, if the objective was to identify an ordering of attributes, they may incur the risk of having to repeat the whole operation again if relations have been mistakenly represented.

It should be appreciated that the approach discussed here only serves to capture the inherent preference structure in a methodical fashion and highlight situations where no such structure exists. It should not be used to force such a structure where one does not naturally exists. This also means that it is not possible to say what type of ordering, if any, will emerge at the end of the analysis process as this, again, is entirely dependent upon the underlying beliefs. To enforce an ordering where one does not exist is to introduce constraints and overheads into operations which are unnecessary. However, if it emerges that only certain relations have been used, with the properties described previously, then it can be inferred that certain types of orderings will apply to the set of preferential relations, as listed in Table 5.15.

Relations Assigned in Matrix	*Order Type*	*Characteristic of Order*
P, nP, I restricted to identical pairs [no J therefore J = {∅}]	Total Order	All elements of *A* are strictly ordered from best to worst with no equivalencies.
P, nP, I [no J therefore J = {∅}]	Total Preorder	All elements of *A* are ordered from best to worst with ties.
P, nP, I restricted to identical pairs, J	Partial Order	Subsets of *A* are ordered from best to worst with no equivalencies.
P, nP, I, J	Partial Preorder	Subsets of *A* are ordered from best to worst with ties.
J	No order	The elements of *A* are all incomparable or there is insufficient data upon which to base a comparison.

Table 5.15: Possible Order Types Emerging from Analysis of Preferences Relating to Attributes

If there are cases of incomparability within the context of expressing preferences, then it should not be assumed that the attributes involved are either equivalent or of no importance to the evaluation as there is also the question of their mandatory nature to consider.

5.5 Scoring Evaluation Coverage

If an evaluation is to be adaptive then it is important to know that the level of service it provides during its modified phases is sufficient to meet the requirements of the consuming process. The objective in being able to describe some notional scoring scheme is to provide a tool by which the level of performance exhibited by the evaluation, either theoretical or actual, can be assessed. Evaluation coverage is defined as the ratio of the actual attributes assessed against the maximum number which could be assessed if the defined attributes were to be employed in full. In considering this issue, the following criteria need to be taken into account:

- Number of attributes
- Frequency of evaluation
- Significance of attributes

The significance of attributes is here interpreted to relate to their preferential ordering, if any is present. In the absence of a preferential ordering, then all unordered attributes are assumed to be of the same significance. Coverage is intended to provide a straightforward interpretation as to the ‘depth’ or ‘richness’ associated with an evaluation and the results it produces. Instead of employing an absolute baseline comparison, which would be very difficult to resolve where matters of quality are concerned, it takes the stance of assessing the current level of evaluation against the maximum level possible under the given definition, that is to say it takes a local view.

This section discusses a mechanism by which the adaptability of an evaluation can be expressed. It does not investigate how knowledge of the scope of such adaptability is obtained in the first instance. The following notation is employed to describe the problem:

E	The evaluation sampling period, in terms of number of instances of evaluation.
p_i	The number of instances of evaluation in sub-period i of the evaluation sampling period, with the sum of $p = E$. Each sub-period is associated with a lower bound on the attribute score level, number of unordered attributes or a mix of both score and number which is required during that sub-period
m	The number of sub-periods within the sampling period.
N	The total number of attributes defined under the evaluation objective.
att_i	Attribute i.
n	The number of unordered attributes employed in an evaluation.
u_i	The number of full unordered attributes used in the sampling sub-period p_i.
d_i	The number of degraded unordered attributes allowed in sampling sub-period p_i.
f	The reduction factor by which to account for degraded unordered samples.
s_i	The individual score of attribute i where the attribute is preferentially ordered.
S_i	The minimum sum of the individual attributes' ranking scores permissible during an evaluation instance in sub-period i where the attributes are preferentially ordered.
C	The coverage score associated with the identified sampling period, reflecting the minimum level of coverage in terms of attributes assessed and frequency of assessment, which can be tolerated.
$maxS$	The maximum score achievable, including both ordered and unordered attributes, if the evaluation was run in full for the whole sampling period.

The assessment of coverage involves the definition of a sampling period identified such that it is representative of, and applicable to, behaviour as a whole and, therefore, its size will be context dependent. If, for example, behaviour is highly cyclic then a precise sample size may be mapped to a single cycle size or length. The sample period may be defined in terms of a number of instances of evaluation or of time. If it is not possible to identify a suitable sample period, for example if behaviour is random in nature, then the different levels of coverage tolerated, instead of being expressed in relation to an identified portion of the sample, might be expressed as a ratio with respect to each other, e.g. tolerated scores of 0.967 and 1.0 maintained at a ratio 1:4 approximately, so that is if there were too many instances of the lower state being run then evaluation should be restricted to the higher value until the performance ratio was brought back into line. This condition is not considered further here.

A dynamic multi-attribute evaluation must expect to encompass attributes which are either differentiated between on the grounds of their relative desirability or are all deemed of an equivalent value. The two situations are discussed below, starting with the undifferentiated case.

5.5.1 Unordered Attributes

It may be that some, or all, of the attributes involved are not thought to make significantly different levels of contributions and are, consequently, all considered to be of the same worth. If this is the case then there is no justification for applying individual scores to attributes and, therefore, coverage must be based upon the number of attributes used instead. The level of coverage can then be obtained by totalling the number of attributes employed within a sub-period of the sample and taking this as a ratio against the total which might have been employed if all the defined attributes had been used continually throughout the sample period,

$$C = \frac{\sum_{i=0}^{m} p_i u_i}{\max S} \qquad [5.1]$$

with $maxS = En$. So, with the sample period defined as being a count of the number of consecutive instances of evaluation, that is $E = 100$ and $n = 5$, then the maximum score which could be achieved if all five attributes were used in every instance of evaluation would be $maxS = 500$. A correct value for C, therefore, falls in the range [0, 1]. If all the five attributes need only be employed for 50% of the sample period and just three used for the remaining 50% of the sample period then C will be (250 + 150) / 500 which yields 0.8.

If substitutions to attribute measures are also to be involved in the evaluation then they too need to be accounted for. However, as there is no ordering involved in this instance they cannot be given a ranking as such. Instead, as each full attribute, when appraised, has a notional value of 1 then a degraded appraisal, as when a substitution is run, can be represented by reducing this value by a constant factor, f, for example $f = 0.5$ reduces the value to 1/2 that of the full attribute. The choice of reduction factor is arbitrary but should be used consistently. Equation [5.1] can then be modified to,

$$C = \frac{\sum_{i=0}^{m} p_i(u_i + fd_i)}{\max S} \qquad [5.2]$$

where $u_i = N - d_i$ So, again for $E = 100$, if three attributes are run at the full level for the whole sample period and two are run at full level for half and are then degraded for the remaining half of the sample, then $C = 0.9$.

5.5.2 Preferentially Ordered Attributes

The assumption is that each attribute has a different preferential significance which needs to be accounted for somehow when assessing the coverage provided by an evaluation. In order to do this, individual preferentially ordered attributes must be assigned a score reflecting their significance or ranked position in the ordering.

Scoring Preferentially Ranked Attributes

Preferential ordering may be arrived at by employing the approach described in Section 5.4 and needs to be translated into a numeric representation using an order preserving function, such that it is possible to uniquely distinguish an attribute's, or a group of attributes', relative position. Given an ordering from the least preferred attribute, identified at the *1*st position, to the most preferred, at the *Nth* position, and an initialising score for the lowest attribute of 1, then the following scoring function is applied which exhibits these order preserving properties,

$$\forall\, j, j \in L \qquad s_i = 2\left(\sum s_j\right) + 1 \qquad [5.3]$$

where L represents the set of all attributes which are preferentially lower than attribute i and thus ranked below it. The individual ordinal scores produced are then normalised, for convenience, to be within the range [0, 1]. Thus, for $N = 5$, the final allocation of individual ranked scores, where all five attributes are ordered and for a_1 is least preferred to a_5 is most preferred, would be as follows: $att_1 = 0.008$, $att_2 = 0.025$, $att_3 = 0.074$, $att_4 = 0.223$ and $att_5 = 0.669$.

As in the non-ordered case, scoring also has to account for those situations where substitutes have been defined, either to stand in for attributes or for their method of appraisal. This can be achieved by considering the substitution to be a *proxy* attribute and inserting it into the ordering between the next lower ordered attribute and the full representation of that attribute with which the substitution is associated. It is then effectively assigned a score which lies between the two. This is a discretionary choice but allows for consistent treatment.

Having assigned individual scores to the preferentially ordered attributes by which to distinguish their contribution, these can then be used in the identification of an overall coverage score with respect to the evaluation.

Identifying the Coverage Score

The objective in providing a coverage score is to provide a means by which the *minimum* number of attributes necessary to the integrity of an evaluation, together with the minimum frequency at which it is performed, can be expressed succinctly. Having assigned individual ranked scores, these can then be substituted into Equation 5.1, as below,

$$C = \frac{\sum_{i=0}^{m} p_i S_i}{\max S} \qquad [5.4]$$

where the maximum score achievable using the scoring function listed here and five attributes is $maxS = 100$, for $E = 100$. If the attributes which must be used during a sub-period contain a mix of both ordered and non-ordered items then both aspects must be combined into one operation, as in ,

$$C = \frac{\sum_{i=0}^{m} \left(p_i u_i + p_i S_i \right)}{\max S} \qquad [5.5]$$

where $maxS = 100 + En$. To illustrate, below is shown the case where five attributes are defined, three of which, a_1, a_2 and a_3, are preferentially ordered and have associated ranking scores of 0.077, 0.231 and 0.692, together with two attributes which are un-ordered. Three sampling sub-periods are identified for a sample period, $E = 100$, covering 70, 20 and 10 instances of evaluation respectively. The minimum number of attributes to be covered for each sub-period is stipulated as follows, parameters being shown in the order of $< p_i, S_i, u_i >$: < 70, 0.923, 2 >, < 20, 1.000, 1 > and < 10, 0.692, 0 >. So, for the sub-period covering 70 instances, ordered attributes to the value of 0.923 must be run as well as both unordered attributes each evaluation instance within the sub-period. The minimum coverage score under this specification, for the whole sample will be,

$$C = \frac{(20*1.000) + (70*0.923) + (10*0.692) + (70*2) + (20*1)}{300}$$

$$C = \frac{251.53}{300} = 0.838$$

It should be understood that, where attributes exhibit a preferential significance, then the coverage score is ordinal in nature, that is it does not indicate the magnitude of performance merely what level in the ordering of attributes has been reached. As a consequence of including the substitution components in the same scaling structure there might actually be listed two or three ways in which an attribute can be assessed and, depending upon the level of the overall score attained, it is to be assumed that only the most important of these has been used in any one instance. Coverage does not expressly represent the mandatory characteristic of an attribute which is to be implicitly presumed from the nature of the specification given. That is to say, the level of coverage expressed should at no point be lower than that which would be achieved by running the mandatory attributes alone, otherwise the two areas of definitions are inconsistent.

5.5.3 Specifying the Evaluation Coverage Required

Having defined a means for assessing the amount of coverage provided by an instance of an evaluation, then the same mechanism can also be used to facilitate the management of adaptive behaviours dynamically by monitoring the ongoing evaluation process and using the information specified in the coverage description to dictate the level of performance required of any one evaluation instance. The following descriptions assume that all attributes are ordered and that there are no substitutions, with $E = 100$ and $N = 5$. Possible ways of employing coverage are described below.

Maximum/Minimum Coverage: By decomposing the sample period into individual units (time or instances) and assigning the distribution of attributes to be assessed across these, then the overall minimum coverage required can be identified which, together with the maximum coverage possible, to be derived from the specification details, provides the upper and lower bounds on evaluation quality.

No Modifications Permitted: If no modification is tolerated then, obviously, all instances and all attributes must be effected in full for the sample period, i.e. $C = maxS$.

Number of Attributes Modifiable: In this case the number of attributes required to be examined must be stated for the appropriate proportion of instances, as shown in the previous illustration, e.g. for the ordered case only and observing $< p_i, S_i >$, < 70, 1>, <20, 0.967> and <10, 0.893> giving 70% of the sampling period where all attributes are used, 20% where the least two preferred attributes can be skipped, and 10% of the sampling period where the lowest three attributes do not need to be assessed.

Number of Instances Modifiable: This case is equivalent to the zero attributes being assessed, e.g. < 80, 1> and <20, 0> means that for 80% of the sampling period all attributes must be used but for 20% of the period it is not necessary to measure any attributes. If this mechanism were being implemented in an on-going monitoring capacity then care must be taken to ensure that skipped evaluation instances are accounted for.

Un-specified Modification: If a lower bound on evaluation is provided and the ways by which this is to be achieved not specified in detail, then it may be possible to realise the modification in several ways. For example, if the lower bound on evaluation is given as C = 0.9, then either of the options below might be a suitable choice:

Option A:	Option B:
<30, 0.967>	<25, 0.967>
<70, 0.893>	<65, 0.893>
	<10, 1>
C = 91.52 / 100 = 0.9152	C = 92.22 / 100 = 0.9222

If both attribute numbers and instances of evaluation, are modifiable then the potential exists for a conflict to arise as, if a number of instances have a requirement to cover a fixed number of attributes per instance, then the instances cannot be reduced and if their number is reduced then the attributes are not assessed. Therefore, the alternative possibilities must be expressed as mutually exclusive options, that is *either option A* is followed *or option B* and once the first modification is undertaken the option it is associated with becomes fixed for that sampling period.

As a digression, a scenario can be considered where an evaluating process might 'lose out' on the amount of service it receives, precisely because it is more flexible in its approach than other processes. That is, because it has named a minimum level of performance below that at which it would ideally like to operate all the time, then this fact is exploited by the system in order to accommodate less 'considerate' processes.

5.6 Identifying Drift

If a dynamic evaluation process is also to be an adaptable one then thought should be given not only to the magnitude of the adaptive range, as characterised in the previous section, but also to the distribution of the different performance levels which is generated as the sample period progresses. In some scenarios it may be thought necessary to ensure that full evaluations are carried out at a certain frequency in order that a complete interrogation of the problem is effected on a regular basis.

It is an assumption that modification is a necessary activity, which can be advantageous in some instances, but that it inherently reduces the 'richness/quality' of the evaluation and should only be pursued where the alternative is no evaluation at all. Modification may be applied to either the number of attributes assessed or the frequency at which evaluation is evoked, resulting in:

- Fluctuations in the appraisal of attributes
- Fluctuations in the inter-evaluation period

The former, gives rise to variations in the depth to which a problem is examined but still allows some knowledge of state to be obtained and, consequently, for values to be updated. The latter prevents any update to information in those instances where an evaluation is skipped. Both result in what might be termed a 'drift' away from the optimal, or full rate, at which the value is refreshed, reflecting the interval between full attribute appraisals.

Uncertainty is a continual risk when attempting to interpret real-world events in real-time. It can be applied to every aspect of operations and is an anathema to the establishment of predictability in behaviours. In the context of this discussion there are two particular areas where it is highly pertinent. The first is the difference between the current results of evaluation, as being employed in subsequent system operations, and the real world state of the evaluated objects. This is an issue which needs to be addressed when interpreting the changeability present in the environment in order to set the evaluation rate at an appropriate level and to decide whether evaluation is dominated by attribute changeability or is process led. The second area is the difference between the current level of performance which the evaluation is operating at and the maximum level it is capable of performing at, given its definition and implementation. This latter aspect is the focus of interest in this section.

Drift is defined, here, as the ratio of the number of degraded samples within an evaluation against the number of full samples. It is, therefore, an extension of the coverage score described in the previous section and is dependent upon the latter being defined and adhered to. However, drift considers the overall impact of degradation as a whole on the sample period, and is not concerned with specific sub-periods. In conjunction with the coverage score, it serves as a tool by which to characterise and specify the degree of adaptability which an evaluation can exhibit.

Drift Value & Usage

Establishing a value for drift can be readily done by taking the ratio of the minimum partial evaluations to the full evaluations, as defined for the sample period under the coverage score. Thus, taking a previous example,

	p	S
i.	25	0.967
ii.	65	0.893
ii.	10	1.0
C = 92.22/100 = 0.922		

where, out of a sample size of 100 instances, 10 are specified to be run at a full level and the remaining 90 at a degraded level. The minimum value for the worst-case drift between instances of degraded and a full evaluation is equivalent to,

$$\textit{Minimum Worst-Case Drift} = \left\lceil \frac{\sum \text{degraded samples}}{\text{full samples}} \right\rceil \qquad [5.6]$$

which for the previous values would be,

$$\textit{Minimum Worst-Case Drift} = \left\lceil \frac{(25+65)}{10} \right\rceil = 9$$

This means that, if the drift is not to exceed 9 at any point within the evaluation sample period then each instance of full evaluation must be followed precisely by 9 instances of degraded evaluation. If the full number of degraded instances is run then this represents the lowest value of worst-case drift which can be consistently achieved across the sampling period. If the drift is seen to be less than this, then it will either be at the expense of a full evaluation being run elsewhere in the sample period or means that a greater number of full evaluations are being performed than the minimum number specified. If this value is considered too large, i.e. it is felt that the interval between running full evaluations is too great, then the number of degraded samples must be revised downwards. For example, the minimum number of full samples required can be established as follows,

$$\textit{Minimum Number of Full Samples Required} = \left\lfloor \frac{\text{Sample Size}}{\text{Required Worst Case Drift Value}} \right\rfloor \qquad [5.7]$$

Within one sample period, if a drift value has not been established as a constraint, then the longest run of degraded instances is equivalent to all of them being run consecutively. However, in reality the maximum worst-case drift which might be observed is equivalent to double the number of degraded samples. This would occur when all full instances are run at the start of the first sampling period followed by the complete number of degraded instances from both groups and then followed by all full instances from the second sampling period. Here, the sample period employs a count of instances and the result is presented as a rational number of no magnitude.

If the active monitoring of drift values and coverage scores is to be pursued on-line then it is dependent upon the capabilities of the evaluating process and system to support it.

5.7 Applying Modification to an Evaluation Process

Enabling multi-criteria evaluation to be adaptable, however, is only one half of the problem. Of equal importance is providing solutions to the questions:

- when to query conditions to find out if modification might be necessary
- when to actually initiate the modification
- what form of modification to employ

with the caveat that system conditions may be stressed at times when these decisions must be made.

Structuring of Evaluation Computation

In the subsequent discussion some assumptions must be made as to how the functionality of the chosen evaluation method is structured. The view taken here is that the instigation and acquisition of attribute measurements precedes, and is separate from, the compilation process. This reflects the belief that, where the primary constraint is one of time, then the former set of actions presents the greater burden and is best executed prior to the point of demand if possible. The notional division of operations, as such, also supports the easier differentiation of the resource budget.

Establishing the Form of Modification

The forms of modification under consideration have been discussed in the Sections 5.1 and 5.2. What form any modification should assume is a question largely resolved by the construction of the attribute scores and coverage scores previously described. If a coverage score is specified this constrains the choice of attributes, and any variants in their behaviour, by indicating the minimum range which must be supported. Thus, the issue of what adaptive choice to make is effectively controlled at the design stage by the range and number of substitutions and variations in behaviours which are listed and captured in the scoring process. However, this still leaves the problem of knowing when to exercise any adaptive capabilities.

Establishing When Modification is Required

The intention is that modification is employed in an attempt to ameliorate any negative impact on performance arising from a named constraint which is assumed to be a physical property of the system, the breaching of which is detrimental to the performance of the evaluation. However, in order to respond to a restriction it is necessary to be aware of it in the first case. The assumption is that the prevailing condition which restricts the evaluation's performance is expressed in the form of a *constraint value* which allows the evaluation to interpret the current state and modify its behaviour accordingly. The point at which the evaluation obtains this information will have a bearing upon what course of action it can subsequently take with respect to applying modifications to its behaviour. Four options are considered as to when to retrieve information about the current state of the constraining property, as illustrated in Figure 5.4. These are listed as follows, together with observations on the implications each has on the ensuing form of any implemented adaptability:

1. *No Query:* no query is made, an evaluation runs until it completes or is terminated whichever falls sooner. This, therefore, requires no form of condition variable or monitoring of constraint issues nor for the evaluating process to be aware of the constraint's condition. It also precludes the use of substitution as a form of modification as substitution is predicated on having prior knowledge of conditions and knowing that such action is required. Modification takes the form of attribute dropping or, if conditions are particularly severe, possibly skipping the evaluation entirely. Whether all attribute measures are received at the same time or evaluation follows a step-by-step approach

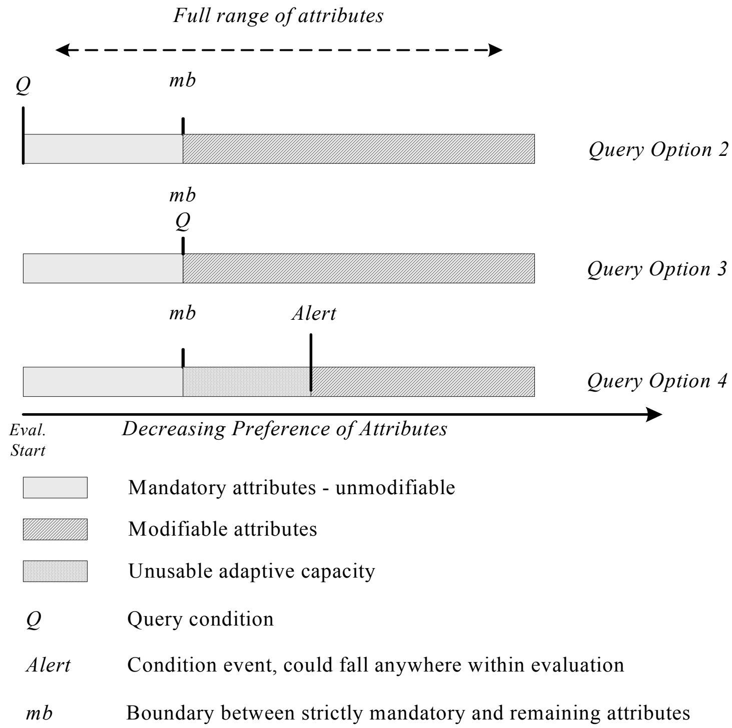


Figure 5.4: Impact of Query Point on Adaptive Range

1. makes little difference. Therefore, the only advantage in applying the coverage score is to observe the level of coverage achieved retrospectively. This option is effectively a default behaviour for when an evaluation process is abruptly terminated or suspended indefinitely.

2. *Query at Start:* the evaluation proactively queries the condition at the start of each evaluation cycle and can, therefore, proceed from the outset knowing what set of behaviours it is to follow. This means that all measurements of attributes can be fetched consecutively and redundant, i.e. non-used information, not accessed. As the evaluator is able to actively identify the current state it can employ all forms of modification should they be available.

3. *Query after Completion of the Mandatory Attributes:* the evaluation queries the condition at the boundary between completing the strictly mandatory attributes (if any) and commencing evaluation of the partially-mandatory/optional group. This leads to effectively the same adaptive options being available as in the previous case only, theoretically, it allows for a fresher view of the constraint state. By definition, no

modification is allowed to mandatory attributes, if they are defined, so in circumstances where they are present adaptation cannot feasibly commence until the group of partial/optional attributes is reached. If no mandatory attributes are defined then this case becomes identical to the previous one.

4. *Alert Update:* the evaluation does not query conditions but receives an alert at any point during the execution of an evaluation cycle. Therefore the data in the condition variable is more up-to-date than the previous case and thus may not incur the pessimism associated with the previous two options. It does mean, however, that evaluation has to be held up while the evaluator responds, conditions are checked and future action decided.

The differences between cases (3) and (4) are that, if the condition variable is defined to return precisely the same value under identical conditions then, in case (3) if the situation is queried as suggested and the interpretation made that conditions are of a certain level then behaviours may be followed which will be undermined if conditions subsequently worsen. In case (4) however, whilst the report is more recent there is a risk of being overrun with updates if the situation is highly dynamic and the evaluation may have already progressed beyond the point at which the level of modification associated with the latest condition value can be applied. Cases (2) and (3) would be of more benefit if the definition of the condition variable were to be more probabilistic in nature.

The above need to be considered in conjunction with the fundamental form of the evaluation, that is whether it follows a breadth or depth-first approach. The latter scenario requires that each alternative object is considered in full before the evaluation proceeds to the next one and, therefore, can only reasonably follow option (2) above if it is to treat each object consistently as a valid evaluation dictates and checks the condition before addressing any alternative. For the depth-first approach following any of the other options may result in the application of modifications which would mean that the same treatments are not employed across all objects, or, indeed that not all objects are considered which may also lead to the results being misinterpreted.

In all cases, the assumption is that attributes are addressed in the order of most preferred to least preferred, with the most preferred attributes being correlative with the mandatory set. Not only does this ensure that, in the event of an evaluation terminating early, the salient, and hopefully sufficient number of, attributes have been addressed it is obviously also required behaviour if modification is to be subsequently applied, specifically in cases (3) and (4) above. Mandatory attributes must be accessed first in order that the optional attributes are still outstanding and, therefore, available for modification if necessary. If the attributes had been handled the other way around, it could be that there were no adaptable attributes left by which to effect some reduction in the evaluation's performance.

A second issue is whether attribute measures are triggered and retrieved concurrently or sequentially. If measurements are obtained, one at a time then this allows for their number to be reduced whilst evaluation is on-going. However, it does carry with it the risk of the evaluation process being held up whilst waiting for updates, specifically in a distributed evaluation scenario. In contrast, if all attribute measures are retrieved at the same point in time, with evaluation waiting till they are all returned, there is little advantage subsequently in deciding to reduce the number of attributes being assessed as what is generally assumed to be the most costly part of the transaction, i.e. networked communications, will have already been instigated.

Form of Constraint Values

Pro-actively identifying a requirement for evaluation modification can only be considered if there is both the ability to characterise the attributes with respect to their impact on the specified constraint and also to relate the interpretation as to the current state with a notion of the budget required to complete evaluation in the various guises. The constraint value associated with each attribute provides the measure of how assessing each attribute impacts upon the specified constraint. It is necessary to define the boundaries by which this value is assessed in order that it is consistently treated. For example, relevant operations to be included in establishing the constraint value in the case where time is the limiting factor might include accessing the attribute at a given location, measurement of the attribute, retrieving the measurement result, processing the result and waiting on the result. A difficulty in this respect, however, is that the constraint value associated with a particular attribute type may not, in reality be uniform across all instances of that type, for example if objects are in different locations. Therefore, in order to generalise the cost associated with processing a particular attribute across all objects it must be decided whether to use:

- a sample mean, on the basis that the individual constraint values and number of objects are knowable and that the difference in value between the mean and the actual performance of attributes exhibiting values which are below the mean level, will compensate for those attributes falling above the mean.

- a worst-case value, again on the assumption that this is a knowable value. Pessimism is inherent in the use of worst-case values but this is the accepted cost of providing some degree of predictability as concern values.

The semantics of the constraint value can support one of two interpretations. It can be defined as either a precise measure of a known physical quantity, which would allow it to reflect exact changes in the real state, or it can be a statistical value, which provides an interpretation as to the probability of certain actions happening, e.g. the probability of system overload. Identifying the form of the latter is a more difficult task which might make for a more pessimistic outlook but encourage more stability in the responses made. However, the alternative to pessimism and probabilistic interpretations is constant monitoring and accurate measurement which are commonly associated with excessive overheads.

*Scored Items**	*Individual Coverage Score*	*Cumulative Coverage Score*	*Individual Constraint Value*	*Cumulative Constraint Value*
a_1	0.64	0.64	20 ms	20 ms
a_2	0.27	0.91	10 ms	30 ms
a_3	0.09	1.00	15 ms	45 ms

*Scored items are attributes, substitute attributes and substitute measurements

Table 5.16: Identifying Modification Level

An example is provided in Table 5.16 to illustrate the relationship between the coverage score, choice of attributes and their variants and the constraint value. If the constraint value is interpreted as supporting 35 ms of work only then by progressing up the ranked attributes the individual overheads associated with each scored item can be accumulated until a cumulative value which is less than, or equal to, the stated constraint is reached. So, using the details shown in Table 5.16, with only 35 ms available only a_1 and a_2 can be accommodated. The inverse situation, where behaviour is led by the coverage score, allows the stated coverage score to be cross-referenced against the impact this will have on the constrained property and indicates the minimum level of support which will be required in this area.

Allowing active consideration of the possible alternatives would be costly to support in terms of overheads and conflict with the notion of a constrained environment. The adaptive range of an evaluation process is dependent upon the number of variants identified at design time. The more alternatives which are included in the coverage scale, then the more flexible and robust the evaluation will be and the broader the range of data which may be handled. Thus, a certain symmetry exists between the inherent adaptive capability of an evaluation process which, in general, might be expected to increase as the size of the evaluation increases, and the greater need for adaptation, which might also be expected to be greater for the more extensive evaluation scenarios.

An important consequence of the above mechanism is that selected behaviours do not necessarily reflect an optimal solution as far as the constrained property is concerned, an approach which can be justified on various grounds. In the context of the evaluation, there are two known causes of evaluation failure. The first of these arises when less than the mandatory set of attributes is assessed, thus giving rise to, what is here termed, a *preferential failure*. The second is that the limitations of the constraint are breached causing a *constraint failure*. From the perspective of the evaluating process these are events of equal and primary significance in the viability of evaluation and are to be avoided where possible. Consequently, any action taken in reducing the risk of one failure should not increase the risk of instigating the other. In addition, the backbone of the adaptive process is the defined preferential ordering to which the constraint issue is anchored, not visa versa, and it is important to maintain the integrity and predictability of the evaluation as a whole by ensuring that it is consistent in its behaviour and not, therefore, switch the key factor driving this behaviour. It might be thought that instead of manipulating attributes on the basis of their preferential order that a solution which is optimal with respect to both the evaluation and constraint could be found. Leaving aside the not insignificant issue of delay in the evaluation and additional computational overheads, it should be realised that under the model described here there is no notion of the *magnitude* associated with an attribute's preference level, or desirability, relative to other attributes only of their respective position in the defined order. Therefore, it is not possible to claim, for example, that $a_1 + a_4 + a_8 = a_2 + a_3$ in terms of the 'amount' of their contribution. Consequently, it cannot be said that, if the first group is 'quicker', i.e. lower constraint values, and evaluation is based on them alone rather than on the second group which take a marginally longer period of time to execute, that the evaluation will be any better off. Alternatively, the dominant ordering could be reversed, that is issues of preference become subordinate to time for a duration whilst the constraint is critical. However, applying a single static ordering to a set of objects on the basis of a finite physical property is not the correct approach when they are deployed in a dynamic context. Rather, the attributes would need to be sorted and re-ordered in the face of each new threat so as to identify the best fit, an action which cannot feasibly be supported.

5.8 Summary

This section has introduced a methodology by which to approach the definition of a dynamic multi-attribute evaluation problem in a consistent fashion, and in such a way that any potential flexibility present within the problem definition can be exploited to enable the on-line evaluation process to exhibit adaptive behaviours. It has also introduced some simple mechanisms by which to facilitate the discussion of the problem and to help manage the adaptive aspects of a dynamic evaluation. The definition and implementation of a dynamic multi-attribute evaluation task is an activity which can be divided broadly into two separate areas of effort. Firstly, the theoretical model by which the problem is to be examined and resolved must be considered in order that the mix of objective and subjective concerns are correctly handled. Secondly, at runtime behaviours must be properly monitored and managed in order that they do not undermine the soundness of the evaluation. Whilst unduly focussing on the theoretical may seem an unnecessary complication to the whole design process, a thorough analysis has to be performed in order to reason effectively about any subsequent adaptation.

Chapter 6

An Illustrative Application of the Framework

Having described a methodology by which to undertake the analysis of a dynamic multi-attribute decision problem, its application to the problem of selecting an appropriate sensor report to pass to a display application is provided by way of an example. The procedures undertaken below follow those outlined in Figure 5.2 and described in Section 5.2.

The problem scenario is that of a sensor-based network deployed in the field to undertake wildlife and habitat monitoring [Aky02]. As such, it draws heavily upon the example featured in [Mai02] which describes a system used to undertake the monitoring of a sea bird colony. Another case is presented in [Ste00] where the environmental observation of a river estuary is undertaken. Indeed, distributed sensor networks, whether localized and small scale or extensive with numerous nodes, can support a variety of customized, data gathering exercises and play increasingly important roles in many areas of research and resource management including traffic planning, meteorology and road safety [RWM05]. Whilst the technologies which underpin such systems have advanced significantly their operation are still not without some difficulty. In addition to the possibility of having to cope with extreme environmental conditions, distributed sensor networks, specifically when using wireless telemetry, have also to contend with tight energy and bandwidth constraints. As the authors in [Mai02] note, careful thought has be to be exercised in deciding how to allocate the, usually finite, energy supply, with several trade-offs to be made between in-network processing, the transmission of raw data, setting of sampling rates and selection of services to run.

6.1 Problem Scenario

Several sub-nets consisting of between 5 and 10 statically located nodes, each of which hosts several sensors, are deployed in geographically distinct locations with the objective of recording environmental data relative to the activities of a certain species of slug. The decision problem to be addressed involves the selecting of one from a batch of the most recent sensor reports received from a specific zone, *zone_1*, the contents of which are then to be forwarded to a web-based application, *display_app*, for display in near-real-time. The basis for selection is not precisely known at the outset, but, as the display application is intended to be part of the public interface with the overall research project, it is desirable that the users have something 'interesting' to look at, wherever possible. It is felt that it is not necessarily desirable to provide detailed graphical representations of each sensor type's activities, or of the activities on all nodes, as this may be too complex and uninteresting when viewed by a casual observer. Also, the drain on resources incurred by handling a large amount of data may be too great. The report to be selected will be one of two whose details are displayed, the other being a fixed choice from a specific location. This latter report is of primary importance with the dynamically selected report being secondary in nature. In the absence of a suitable report being identified then either library data can be used or the selection can cycle through sensors on the basis of their location. There are several reasons why the evaluation may fail, for example as a consequence of network or node failure, or the particular group of sensors being powered down at the time the user logs on or, indeed, none of the reports being considered as sufficiently interesting. Thus alternative behaviours are acceptable.

Each sensor is identical in its structure and in the range of functions it supports, which include basic signal processing, analogue to digital conversion of signals (16 bit data types) and local management functions, including monitoring of defined signal thresholds. In order to conserve energy, some of the sensors, namely barometric pressure, relative humidity and temperature, are activated, or deactivated, only when a specified threshold has been reached. As a consequence of this, the amount of data generated each cycle will be dependent upon which sensors have been active and for how long. Communication is via wireless, using a low power time division multiple access (TDMA) based MAC protocol, such as that discussed in [Leg04], allowing transmission at a rate of 40 Kbps. Each node directly communicates with the base station in a single hop. In addition to acting as the communication link, forwarding data from the sub-net to the external network, the base station hosts management functions which co-ordinate the nodes' behaviour. It also performs the data acquisition function, receiving the reports from individual sensors, processing the data and then storing it for access at a later date. Owing to its central role in data handling, it is intended that the base station also host the selection function by which to identify a suitable report for display, and forward a copy of the chosen data onward directly to the display application. Interaction between nodes and base station is cyclical in nature giving each node a fixed slot of 4 seconds in which to transmit its current report. The base station operates on a cycle of 60 seconds, collecting a report from each node in turn. This leaves it with 20 seconds in which to undertake the data processing and archiving functions together with the range of housekeeping tasks, e.g. health monitoring and uploading network control signals and interfacing with external entities. If each node has returned the maximum data then the subsequent processing time required will eat into the slot available for domestic activities, which will include the evaluation task. As maintaining network and node functionality is of greater importance than fulfilling the display role, the evaluation task will be delayed or indefinitely postponed if there is a severe backlog of work. Each report transmitted from a sensor node contains a summary of the activities of the last 60 seconds. The details of sensor types and their summarized data entry in the report are given in Table 6.1. The thermopile is a passive infrared sensor (PIR) used to detect the heat from black bodies in the sensor's field of view. The thermistor employs the temperature difference detected by the thermopile to work out the temperature of any such body.

Sensor Type	*Sensor Sample Rate*	*Digital Output Range & Unit*	*Representation in Summary Report*	*Maximum Entry Size Per Summary Report (bits)*
Humidity*	500 Hz	0% - 100% relative humidity	Per Second Mean	960
Barometric Pressure*	10 Hz	300 – 1100 mb	Full Sample	9600
Thermopile	2000 Hz	0^o - 40^o C temperature difference	Per Second Mean	960
Thermistor	2000 Hz	(assists thermopile functions)	n/a	n/a
Temperature*	2 Hz	-5^o – 50^o C	Full Sample	1920
Photoresistor	2000 Hz	0 - 50,000 lux	Per Second Mean	960
Maximum Total Summary Report Size (bits): 14,400				
* Indicates sensor activity is threshold activated.				

Table 6.1: Sensor Types and Summary Report Entry Details

In addition to the report on sensor activity a node can also forward a single image each cycle. This is triggered by a positive PIR reading, above a stated margin, which may indicate the presence of an animal within the sensor's field of view, and will continue to be taken on a minute-by-minute basis whilst the state persists. The signal to activate the image sensor is initiated from the base station once it has identified the necessary conditions exist and, consequently, there will be a small lag between an entity being within range of a sensor and the first image being taken. Likewise, the final image in a series might be executed after the condition no longer exists if it is run before the node receives the termination signal. Images assume a 200 x 150 8 bit greyscale JPEG format, with compression at 50%, making the total size approximately 15 KB. This makes the maximum amount of data which the base station might expect from each node, each cycle, 16.8 KB excluding packet overheads.

As indicated, it is expected that the evaluation task will be executed locally on the base station after the current batch of reports and images has been received. As a result of performing the evaluation, the base station will then copy the most appropriate report/image identified and forward to the *display_app* which has the responsibility for creating and managing the display itself.

6.2 Worked Application of Evaluation Analysis

Below is a 'walkthrough' of the analysis framework, as applied by a single decision maker to the previous problem. The intention is to demonstrate how considering the nature of the decision problem and its treatment in a holistic fashion, whilst at the same time exploiting any adaptive capacity present in the problem description, leads to a more integrated and effective design process.

6.2.1 Section One - Defining the Problem

Section One addresses the question of whether the identified decision problem is of a dynamic nature and what the structure of the problem is.

A - Outline Description of Evaluation: In this instance, the previous description of the problem scenario can be assumed to serve as an outline description of the evaluation problem. Brevity in producing such descriptions is to be preferred whilst the description remains sufficiently detailed to provide a unique description of the problem and characterise the high-level elements of the same. In short, it serves to provide a functional specification of the problem including details of the issues of concern, together with known constraints and components of relevance to the problem.

B - Validation Check: An essential question to ask before proceeding any further is whether the problem, as described so far, warrants dynamic resolution. Therefore, applying [Q1] to this particular problem description it can be seen that the issues of concern are not knowable off-line and that, whilst the alternatives may be identified beforehand, nothing can be inferred about the actual state of any one report as this is dependent upon the external environment which is beyond system control. Given two negative responses, then it is reasonable to assume that the problem would warrant dynamic investigation.

Of course, it might be possible at this point to redefine the problem into a static format and, thus, do away with the need for a dynamic treatment, for example deciding to employ on the known location of each node as a the sole criteria for selection. However, in this instance, given the dependency on external, unknown conditions, then any solution which wishes to account for such conditions has no option but to take a dynamic approach.

C - Complete Set-up Process: Having been satisfied as to the need for dynamic evaluation, the next stage is to complete the problem definition and characterisation in order that consideration can be given to the nature of the problem in more detail.

C.1 - Definition of Terms: A hindrance to progress in any activity is a confusion over terminology employed, therefore before proceeding any ambiguous or unfamiliar, problem specific terms, should be clarified and common definitions noted. It is also necessary to clarify which elements are playing what role within the problem. For this problem the roles are allocated as follows: the decision maker - software engineer (named); the evaluation host process - to be located on the base station and identified as *Proc_1¸* attribute update process - this role remains vacant at this point as the attributes and other behaviours relating to evaluation, have yet to be identified; consumer - also *Proc_1*, making both the act of evaluation and consumption of results internal to the same process.

There could be many variations in the responses under this section, for example there could be several bodies involved in the decision making role, including an unknown end-user. Likewise, the evaluator's host, attribute manager and consumer might all be different processes. It is necessary to identify the correct group of decision makers so as to pin-down precisely whose opinion must be sort and accounted for. From the perspective of an evaluation, it makes little difference whether all procedures are internal to the same process or handled separately so long as the activities are satisfactorily synchronized and communication between processes not hindered. However, in reality the management of process interaction must allow for potential delays and synchronization difficulties between the various components.

C.2 - Define Evaluation Objective: The definition of the evaluation objective, together with attributes and object definitions, constitutes the main body of the problem detail. The criteria are not listed in the strict order of completion and it may be necessary to postpone the descriptions of some items until others have been addressed. The objective definition should contain sufficient information, in a concise format, as is relevant to the exploration of the decision problem, rather than excessive extraneous material. Where assumptions are made in the absence of insufficient data, this fact should be noted.

Table 6.2 shows the completed definition for this problem. Insufficient data may be available to complete all elements at this point, for example item (viii), and these must be returned to later. For the alternative objects, a general description of the type is required, together with specific descriptors of individual instances, if known, as given in Table 6.3, though in this case only one specific instance is shown to save space (items ii and iii).

Before the definitions of the attributes to be used can be completed further thought must be applied to the problem objective and any specific criteria mentioned in either the problem scenario or objective definition. In some situations the issues to be represented may be precisely given. Here, however, the specification, whilst it has listed some factors of interest, has left the final choice open to the software engineer to identify and justify. Subsequent testing may prove that the attributes named here are not, in fact, the most suitable and adjustments may have to be made and noted.

The objective definition lists some factors of interest from which appropriate attributes might be drawn. In addition, there are the individual discrete data produced by each sensor. By considering the options available and the usage to which the *display_app* intends to put the data it should be possible to refine the number of possibilities. It is known that there may be some temporal constraints so it is reasonable to suppose that it would be advantageous to keep the number of attributes to a succinct number. Also, as criticality is not an issue directly, then it may be feasible to perhaps consider some less obvious choices which might have a certain risk associated with them.

	Criteria	*Description*	*Options*
i.	Definition	Objective: To select single report from current batch on the basis of its content and condition, data contained to be used as part of promotional/informative application (*display_app)*. Source of reports limited to *zone_1*. Basis for selection: Open - factors of potential interest include age of report, individual sensor readings, health status of node, location of node, presence of target fauna. Means: State of reports to be interpreted from appraisal of attributes reflecting factors of interest. Execution of evaluation is tied in to cycle of report retrieval and processing giving maximum of 1 evaluation/per cycle. Form of Solution: To be established. Context: Identified report required as input to graphical & image display application (*display_app*); in conjunction with primary report display. The functionality of *display-app* is dependent upon report update rate and evaluation rate and is not presumed to have any input into establishing evaluation rate. *Zone_1* may be intermittently inactive. Nodes sited at fixed locations around zone; locations in order of proximity to perceived centre of activity - (a, b), (c), (d), (e, f, g, h), (i, j). Rationale: Desire to make display dynamically responsive to monitored conditions; uncertain as to whether single specific dynamic condition sufficiently frequent or relevant to make single criterion evaluation so assumption that if consider several factors may getter more interesting level of response.	
ii.	Satisfaction	The evaluation produces a usable and consistent set of results within the time available such that it is possible to identify the single alternative which best represents issues identified; to select single best available alternative. In absence of evaluation failing to identify acceptable report either through failure to execute of insufficient activity, default to selecting report on random basis. Evaluation to observe any coverage and drift requirements if identified.	
iii.	Size of Problem	Number of objects: min = 0; max = 10; (complete failure of all nodes - all nodes functional). Number of attributes: To be established, preferably ≤ 5.	
iv.	Criticality	The inference from the problem description is that the secondary report is desirable but functionality can be maintained, albeit at a reduced level, without it.	Non-critical
v.	Frequency	1 per cycle of report processing. +/- *n ms* owing to fluctuations in base station workload.	Repeat

Table 6.2: Defined Evaluation Objective for Sensor Report Problem Continued…

vi.	Type	Reduction	
vii.	Comparative Form	Local	
viii.	Value Uniqueness	tbc	
ix.	Alternative Type	Described in Table 6.3.	Max 10
x.	Attributes	Described in Table 6.4.	5
xi.	Constraints	Time constraints on *proc_1* relating to communications cycle; proposed fixed communication slots with nodes of 4 seconds each, cycle of 60 seconds. Constraints relating to factors of interest not known.	Level 3

Table 6.2: Defined Evaluation Objective for Sensor Report Problem

	Criteria	*Description*	*Options*
i.	General Description	Per node data report on sensor activities. Frequency of report controlled by data acquisition rate. Node architecture and sensors suite of type (*named type)*. Scope of interest confined to those deployed in *zone_1*.	
Specific Entry Example:			
ii.	ID	Z1.1	
iii.	Spec-Z1.1	Location: Site Z1.a.	

Table 6.3: Defined Alternative Objects' Criteria for Sensor Report Problem

The immediate choice would be to extract each individual sensor's data as an attribute and establish the relative merits of the overall report on the basis of the individual values. This would make the end display directly reflect the prevailing environmental conditions as perceived by a particular node. However, on further consideration, there may be certain difficulties involved in pursuing this approach partly arising from the context of the monitoring and partly from the specification of sensor operations. The purpose of the research in which the sensors are employed, is to identify those sets of environmental conditions which are relevant to the behaviour of interest in the target fauna. At this point, prior to the research being performed, these conditions are not known precisely, therefore assuming that, for example high relative humidity is of more value than high temperature has no justified foundation and may not, in terms of the final display, lead to an interesting graphic/image presentation. Also, as some of the sensor types are threshold activated, so as to conserve energy, then there is no knowing whether the threshold has been set at the right level, i.e. ties in with conditions of interest, or whether the sensor will be dormant or active at a particular point in time. If composite properties are inferred from several sensor inputs these may be produced insufficiently frequently to form the basis of an interesting graphical display. Finally, from the perspective of a non-specialist member of the public, what is of interest to them may not be what is of interest to the researcher. Another factor is that, a sensor node may be subject to re-tasking such that the interaction and functional role of its constituent sensors may be adjusted, again leading to inconsistency and possible difficulty if attributes are tied to a particular tasking format.

For these reasons, basing evaluation exclusively on data content elements may not be an easy or, necessarily, suitable approach to take. Therefore, it might be of interest to consider two other sources of attributes, which might produce attributes of a more generalised and less detailed nature. Firstly, there are what might be termed *meta-attributes*, that is descriptors which characterise an individual node's sensor report as a whole. Examples of such attributes might be timestamps, content size, and node location. A second source might be derived from physical properties of the node itself, of which attributes reflecting energy and communication activities may be of particular interest.

Considering all these factors, the software engineer, in his role of decision maker, must identify a set of attributes to work with. Here, the choice made at this point is presented in Table 6.4, together with justifications. Each attribute should merit its own entry and description and the significance of an entry should not be inferred from the order in which they are entered.

The identifier (ID) should assume the common format and be consistent with other cases. The definitions outline the attribute and give the decision maker's justifications for his choice whilst the ‘owner’ category is intended to locate the attribute within the system context. As regards the question of criticality, the problem description does not state precisely that the evaluation is mandatory therefore it may be presumed that it is optional and, if it is optional then no attributes, under the definition, can assume a mandatory form. Whilst this increases the flexibility associated with the evaluation it also, potentially, increases the amount of work which might be required on the part of whoever has to implement it. Conversely, a strictly mandatory evaluation reduces the number of opportunities for response but also simplifies the implementation process. The detail entailed in identifying the type of value extraction will be dependent upon what is known upon the precise system and owner specification to-date. It should, however, be sufficient to avoid any ambiguity. In considering the question of whether the issue represented by the proposed attribute is tolerant of being traded-off against all or any other of the attributes, the decision maker must use his judgement and expertise to arrive at a

	Criteria	*Description*	*Options*
Attribute 1			
i.	ID	Z1.proximity	
ii.	Definition	Description: Distance of node from current perceived centre of activity in monitored area; semi-permanent value – the nodes are statically located but it is uncertain whether the centre of activity might change once information about the area has been received. Issue Representing: Proximity of node which submitted report to possible activity; decision maker's assumption that location may be factor in monitoring greater or more interesting activities for display purposes. Justification for Choice: From the perspective of the evaluation assumption that nodes sited closer to centre of activity may by default generate more activity.	
iii.	Owner	Node	
iv.	Criticality		optional
v.	State Appraisal	Copied from transmission IDs; 4 bit if precisely 10 locations; closer proximity - higher worth.	integer ID
vi.	Trade-off Tolerance		tolerant
vii.	Value Dependency	The value is independent of other potential attributes.	independent
Attribute 2			
i.	ID	Z1.energy	
ii.	Definition	Description: Node assessed report of remaining battery life. Issue Representing: Node's remaining energy supply. Justification for Choice: If energy levels are low then there is possibility that either node will cease transmission permanently or, shut down for this duty cycle, whilst recharges. Either case may lead to the abrupt dropping/substitution of a potentially interesting display when it fails to submit a report in the next batch. This may lead observers to query the soundness/integrity of the research project.	
iii.	Owner	Node	
iv.	Criticality		optional
v.	State Appraisal	Node assigned reading of current battery voltage level; 8 bits; larger value - higher worth.	
vi.	Trade-off Tolerance		tolerant
vii.	Value Dependency	The value is uninfluenced by all other attributes.	independent

Table 6.4: Defined Attribute Criteria for Sensor Report Problem Continued ...

	Criteria	*Description*	*Options*
Attribute 3			
i.	ID	Z1.report_size	
ii.	Definition	Description: Size of the total report submitted by a node, excluding image. Issue Representing: Indicator of node activity and implied richness of data content. Justification for Choice: Handling specific sensor data is complex task, e.g. do take raw data or processed version, for reasons sited elsewhere. Given known fact that a known proportion of sensors are threshold activated and that if they are inactive they do not submit a report, then the inference is that more reports mean more activity and visa versa. Taken as simple way of representing general content state.	
iii.	Owner	Summary report	
iv.	Criticality		optional
v.	State Appraisal	16 bit value; measure in bits; min value 1920; max value 14,400; larger size - higher worth.	
vi.	Trade-off Tolerance		tolerant
vii.	Value Dependency	The value is independent of other potential attributes.	independent
Attribute 4			
i.	ID	Z1.image	
ii.	Definition	Description: Per sample cycle single still image of sensor field of view. Issue Representing: Provision of interesting display material. Justification of Choice: Images are provably more readily understood than displays of raw data or pure graphical interpretations. Presence of an image, particularly one capturing some out-the-norm activity, will be more appealing to observers and will also help in interpretation of graphical presentations, if any.	
iii.	Owner	Node/Summary report	
iv.	Criticality		optional
v	State Appraisal	Image file size > 0; binary flag on image present yes/no; 4 bit; yes > no.	
vi.	Trade-off Tolerance		tolerant
vii.	Value Dependency	The value is independent of other potential attributes.	independent

Table 6.4: Defined Attribute Criteria for Sensor Report Problem Continued ...

	Criteria	*Description*	*Options*
Attribute 5			
i.	ID	Z1.presence	
ii.	Definition	Description: Passive infrared temperature difference from sensor's field of view. Issue Representing: Provision of interesting display material. Justification of Choice: Positive difference value > ambient temperature may indicate presence of fauna and will trigger image capture. Also, zero energy consumption and continual activation gives stable, continuous signal which is useful for display.	
iii.	Owner	Summary report.	
iv.	Criticality		optional
v.	State Appraisal	Sensor return; + value difference; units 0^0 C; worth increases with value; 16 bits.	
vi.	Trade-off Tolerance		tolerant
vii.	Value Dependency	The value is independent of other potential attributes.	independent

Table 6.4: Defined Attribute Criteria for Sensor Report Problem

conclusion. A simple approach is to ask himself whether he would consider trading some quantity of the attribute he is in the process of defining for any, or all, of the others. If the answer to this is yes, then he might assume that he will be able to trade-off this attribute subsequently (undertaking this process therefore presupposes knowledge of at least some of the other attributes, so it may be that the completion of this criteria is left until all attributes have been identified). This is a highly subjective issue, particularly where behaviours are not driven by purely functional requirements. Likewise, with the question of value dependency where trying to accurately disentangle the assumed relationships between attributes may require some careful pondering.

Whilst the identification and definition of attributes is a subject in itself and not the intended focus of activity here, it should be noted that the definition processes should concentrate on the structuring of the evaluation and the identification of intrinsic issues of concern, for example discrete criteria of concern, and not seek to provide value-based interpretations ahead of them actually being identified. Therefore, it should not presume to explicitly state the relative worth of attributes as establishing such information is an act more applicable to the point at which the chosen evaluation method is defined.

Considering the criteria of owner, it can be seen that attributes are either associated with the nodes directly or with the summary report. This would suggest that the responsibility of managing attribute updates is divided, as appropriate, between the individual node (for proximity and energy) and *Proc_1* (for report_size and presence, and possibly image as this latter criterion may be better handled here than by the node itself).

C3 - Identify the Evaluation Rate: Having completed definitions and descriptions as far possible at this stage, the next step is to consider what might be a suitable rate at which to conduct the evaluation. In this case, this factor is already known as the problem description clearly states that a maximum of one evaluation is expected to be undertaken after the retrieval of each batch of reports and, as in this case it is known that report production is tied in to a 60 second cycle, then a precise frequency is known. It is also apparent from the description that the cyclical nature of data production would be the dominant factor regardless of whether it had been stated explicitly or not. That is to say, the *webb_app*, which it must be remembered, although it is not the direct consumer of the evaluation results must handle the consequence of the evaluation process, has not been identified as having any influence in the frequency at which the data is examined. However, it may be that the description is not quite so forthcoming in other situations or that there are conflicts between a perceived 'convenient' evaluation rate and the rate at which, say, the consuming process wishes a result to be produced. In such cases there are trade-offs to be identified between a desirable level of evaluation and a level which can be realistically supported.

Some insight into this question can be gained by considering the issues raised in Section 5.1.5. The first step is to identify which of the particular changeability scenarios applies to the problem environment. In this case it is Item (4) in the list given in the section, i.e. that of fixed alternative objects and dynamic attributes. Intentional mobility is not part of the sensor nodes' functionality with the alternatives remaining constant whilst they continue to be functional. As a consequence of this knowledge, it is readily apparent that attribute dynamism must be addressed directly and the next step then becomes one of deciding which of the four options might be the most suitable approach to take in order to manage the inherent complexity of the problem. Here, network constraints preclude heavy network traffic, e.g. intensive querying of individual nodes as would be required if attributes were to address issues directly below the level of summary report, that is before data had been transmitted to the base station. As the base station co-ordinates operations on the 60 seconds cycle then it would appear appropriate to define attributes which can reflect this rate of change, that is above the level of fine granularity, highly dynamic activities as reflected in the individual sensors' sampling rates. This makes Option I redundant. A statistical approach might be suitable, but on further consideration there is insufficient knowledge both of the lifetime and monitoring operations to ascertain whether there is enough data, and sufficient variation within that which there is, to ensure an interesting presentation, so Option IV might also be passed over. The pessimistic approach, Option III, too, is not really applicable in this case as the only potentially known constraint is that of time, as applied to the operations of *Proc_1*, which, if enhanced by a pessimistic interpretation will have the result in reducing runtime flexibility even further. Also, the evaluation objective has identified the problem as being of a local type, so global constraints are not expected to be employed. Therefore, by a somewhat ad hoc process of elimination, the only choice remaining, which may prove to be the most suitable, is that of fixing the level of attribute definitions such that they address issues which exhibit a level of dynamism which it is considered that the system can support, that is Option II. In a real situation, redefining the attributes so that they operate at a manageable level of dynamism may be the most frequently referenced solution but there is no guarantee that any of the four complexity reduction heuristics will be entirely suitable.

Consideration of the evaluation rate is undertaken after the attribute and objective definitions have been provided as the integrity of the decision resolution process depends, in part, upon employing what the decision maker considers to be the most appropriate set of attributes and not on picking arbitrary attributes because they can be more conveniently manipulated. In this example, the defined attributes already reflect the Option II approach as they take advantage of the functionality of the underlying processes, specifically the summary characterisation. If they had been targeted at different levels of activities, then their definition would have to be revisited and possible composite/proxy solutions arrived at. An indication of what frequency of evaluation is to be expected can be derived from acknowledging whether the whole operation is to be driven by the inherent changeability of the phenomena which the attributes address, or by the process which consumes the results of an evaluation. If the former were to be the case then the dynamism of the set of attributes would require modelling and the evaluation tasks and usage of evaluation results structured around a changeability level identified as appropriate. If the latter situation applies, and evaluation is process driven, then it remains to consider which of the two key processes involved, the evaluator and the consumer, plays the dominant role and structure the subsequent evaluation around the relevant set of activities.

It may be that addressing the question of what is an appropriate evaluation rate might benefit from further analysis, for example employing the *pseudo-probability* scheme described in Section 5.3. This, to recap, is based on the assumption that for a preferentially ordered set of attributes, regardless of the exact form of the methodology chosen to perform the evaluation, the more preferred the attribute then the greater will be its influence in the final result. Therefore, by considering a particular level of change, for example the change of all mandatory attributes across all objects, one can identify the appropriate pseudo-probability by which to describe this level. In conjunction with any assumptions and known characteristics about the dynamism of the individual attributes, the attribute/object interaction might be simulated to see what effect this level of response might have upon the evaluation's performance.

The difficulty for an evaluation activity which is to be attribute, rather than process, driven is that the individual change rates of attributes may be completely at odds with their preferential significance. As a consequence, any evaluation set to respond to attributes which displayed a certain rate of change and higher, might find that these are actually of a low preferential order and, therefore, their influence within the re-evaluation is minimal, with the probability of the alternative objects being reordered as a consequence lower. If an evaluation is of a repeat type then this may result in an unreasonably high rate of work when compared to the differences observed between sets of results. A crude rule of thumb (which remains to be explored) might be that, for attribute-led evaluation, it may be applicable to identify the high preferentially ordered attributes and focus the rate of evaluation upon their change rate, e.g. if it is low then evaluation is run at a low rate and visa versa, on the assumption that there is a higher probability of any re-evaluation under this scenario bringing about a re-ordering of the set of alternative objects. However, as has already been stated, the actual variation in the results between evaluation instances is a consequence of the interaction between the number of objects, the level of precision at which the values are represented, the range at which they are represented and the differential treatment of attributes expressed in the evaluation methodology. Therefore, trying to establish the balance between the appropriate evaluation workload and the inherent changeability of a set of attributes and the issues they represent, would be far from easy.

At this juncture in the process of evaluation analysis, however, no preferential expressions have been provided. Therefore, if the pseudo-probability approach is to be used it must be postponed until later, or such information provided here. Pseudo-probability can be adopted in the absence of preferential ordering, by substituting the number of attributes changed rather than their order and only addresses known dynamic attributes. Therefore, in this case as the proximity attribute is effectively to be interpreted as being static to all intents and purposes, only the remaining four attributes should be considered. Energy is defined precisely to represent the power remaining at an instance immediately prior to each summary report being transmitted and, therefore, its rate of change is tied into the production cycle for summary reports. This is also the case with report_size and image where the attribute is defined in the context of summary report production. In contrast, presence is statistical in nature and, as such, is dependent upon its definition to identify the dynamism expected of it. Here, as with the other attributes, it is tied into the report cycle. The evaluation, as driven by *Proc_1*, reflects a pseudo-probability of 1.0, in that all dynamic attributes are assessed on all alternative objects and all are expected to have changed their state under each evaluation instance.

Finally, with reference to changes in values, it should be noted that such changes may not necessarily reflect great differences in the actual state of the alternatives which they are representing. If the level of precision is high but the range of values is small, it can be anticipated that the tendency will be for alternatives to be distinguished effectively as a consequence of the precision applied rather than significant differences in state. Alternatively, if the precision levels are small and value ranges high (reflecting an expected real world distribution of values) then the reverse might be expected, that is alternatives would be distinguished rather by differences of state than as consequence of the supported precision levels.

This concludes the stages required under Section One. Before proceeding, incomplete entries should be revisited in order to see whether the missing information has been acquired in the meantime. In this instance the data uniqueness section of the evaluation objective definition could not be completed earlier owing to insufficient knowledge of the attribute types. Now, however, working with the definitions proposed it is possible to gain an idea about whether the uniqueness factor associated with a set of results is sufficient in itself, and independent of the real state of the attributes, to generate a satisfactory outcome given the type of evaluation solution required. In this case, that means considering whether the likely uniqueness will be sufficient to give rise to a single outcome, attribute states permitting. As there are only 10 alternatives, and five attributes, two of which at least it is proposed be of 16 bits, then the number of feasible value permutations can be seen to be far greater than the number of alternatives. This means that, should equivalencies occur, then they are a better representation of the real state of the alternatives than if the precision of the values were lower and real conditions more broadly categorized. Relative to the number of alternatives, Criterion viii in Table 6.2 can now be identified as being *high*.

6.2.2 Section Two - Considering Adaptive Capabilities

Having been satisfied that a dynamic approach to the problem resolution is suitable, Section Two goes on to address the question of whether it is desirable and, indeed, feasible that the dynamic evaluation also be adaptable.

D - Are There Constraints on Operations? The initial question in this part is straightforward. If performance is not to be hampered by constraints relating either to the system resources or traits associated with the alternatives themselves, then there is no need to consider the question of adaptation further. The analysis can move on to step D1 and apply the information gathered so far to the task of selecting an appropriate evaluation method. If, however, there are known certain, or potential, constraints, then it is worth proceeding to see if an adaptive response might form an adequate solution to these. In this example there are stated constraints relating to the performance of *Proc_1* which is expected to host the evaluation process and these are expected to be of a temporal kind.

E & F - Check Constraints & Consider if Modification is Advantageous: From the general description of the problem scenario and the defined evaluation objective, specifically the description of the stated constraints (Table 6.2, item xi), it is impossible to draw any conclusions as to whether the constraints are going to be a frequently encountered problem or not. The evaluation has been categorised as non-critical, so in the absence of any further knowledge, the decision maker might opt to ignore the potential impact of constraints, considering there to be an insufficiency of information so as to warrant any further effort in this area. It might be, however, that previous experience with similar sensor-based operations, or a knowledge of the environmental conditions which are likely to be encountered, has provided the decision maker with an insight into the fluctuation in system workload likely to be encountered between sampling periods and led him to conclude that, in the absence of any evidence to the contrary, he should take a pessimistic view and assume that there is a possibility of there being extended periods where there would be insufficient time left to complete the full evaluation, as described so far. The principal concerns with respect to the way that constraints, particularly temporal ones, can impact upon a dynamic evaluation, especially where it is repeated as is this one, are that they either prevent it being run at all or they persistently prevent it being run in full. Either of these two options, extended over a continuous period of time, may give rise to undesirable consequences elsewhere in the system. In the absence of details of process interactions, the decision maker must apply an informal 'guesstimation' as to what the most commonly encountered problems are likely to be and whether they can be countered by modifying the evaluation. In this case, it is known that *Proc_1* has been allocated 20 seconds to perform a wide range of tasks. Some of these are known, such as processing the received data, whilst others are not, particularly network related activities which may fluctuate widely. Therefore, in this instance the decision maker might be justified in assuming that, in most instances, there will be ample time to complete a substantial proportion of the defined evaluation, but there may be rare occasions when it cannot be run at all. Therefore, and given the publicity factor relating to the intended usage, it is valid to consider possible adaptive approaches if available, providing a solution to step F.

If the response to step F were to be in the negative, then the rest of Section Two might be skipped and the decision maker proceed directly to considering the choice of evaluation method. Before this, however, it may be advantageous to revisit Section One, and revise the evaluation definitions in order to permanently account for the potential constraint. The ways in which this is achieved are context dependent. Here, possibilities include reducing the number of attributes.

G - Check Criticality Types & Get Modification Range: So, having decided that being able to modify the evaluation process dynamically might be beneficial it must be shown that the process can actually accommodate such an approach. Asking Question 3 from Section 5.2.2 is a starting point in this area with the answer, here, being [no, no, yes] with respect to the three sub-queries. This then opens up the possibility that modification can be tolerated and that it may consider taking the form of skipping an evaluation instance and the dropping of attributes. In fact, it is known from the attribute definitions that they are all optional and, in conjunction with the fact the evaluation objective is identified as being non-critical, then with reference to Table 5.8 in Section Two, it can be seen that all four types of adaptive response are allowable.

H - Is Modification Feasible? Following directly from the previous point, it then follows that in this case the answer to this question is yes. If the answer had been no, then the analysis would have continued with Section Three and the choice of methodology, though again not before possibly revising the definitions so as to make the evaluation more robust in face of the stated constraints. Consideration then needs to be given to whether either attribute truncation might be employed or whether adaptation is to make use of the coverage/drift rate mechanism. If neither of these is required, then the analysis process can skip these stages and proceed to Section Three. However, here the intention is to apply both mechanisms in order to better manage adaptive behaviours.

J - Define Preferential Order: In the example given here, there are only five attributes employed which have been defined by the decision maker, theoretically on the basis of representing the most useful criteria by which to investigate and resolve the stated decision problem. However, if the dynamic resolution of the problem is to be achieved then the theory has to be instantiated in a physical form and, in this case, the decision maker also assumes the role of implementer as well. Therefore, it should be acknowledged that, subconsciously, additional factors have influenced his choice of attributes, for example as to how he envisages they might be employed. As a result, he may already hold a very strong opinion about the relative significance of each attribute, particularly as only a few are defined. Alternatively, he may have thought about their collective usage and not considered them to be individually distinctive.

If the system is to manage multiple attributes in a structured manner then it needs some guidance, or rules base, to assist it in the task. The belief in this work is that expressions of preference are a valid means to facilitate the differentiation between attributes in a decision scenario. This does not preclude the consideration of other factors, particularly if the practicalities of implementation are to assume a greater prominence, however they are not dealt with here. Assuming that, in this instance, the decision maker is uncertain as to what his preferences with respect to the set of attributes are, he employs the preferential ordering scheme described in Section 5.4, to help resolve his opinions.

Table 6.5 shows his completed preferences and the identified final total order resulting from a count of each sensor's awarded relations. In order to aid himself in identifying the relations, the decision maker follows the procedure described in Section 5.4 and identifies the relevant relations from rows already completed in Table 6.5. These relations are shown in Table 6.6. From these, and in conjunction with the details given in Table 5.9, the available permissible relations are shown in Table 6.7. Pairings to the left of the diagonal do not need to be considered directly as they can be inferred from the remaining entries.

	Proximity	*Image*	*Report_Size*	*Presence*	*Energy*	*P Count*	*nP Count*
1. *Location*	X	nP	nP	nP	nP	0	4
2. *Image*	P	X	P	nP	P	3	1
3. *Report_Size*	P	nP	X	nP	P	2	2
4. *Presence*	P	P	P	X	P	4	0
5. *Energy*	P	nP	nP	nP	X	1	3
Order of Preference for Attributes: Presence *P* Image *P* Report_Size *P* Energy							

Table 6.5: The Decision Maker's Expressed Preferences between Pairs of Attributes

Row	*Proximity*	*Image*	*Report_Size*	*Presence*	*Energy*
1. *Location*	X	n/a	n/a	n/a	n/a
2. *Image*	-	X	1. < nP, nP >	1. < nP, nP >	1. < nP, nP >
3. *Report_Size*	-	-	X	1. < nP, nP > 2. < P, nP >	1. < nP, nP > 2. < P, P >
4. *Presence*	-	-	-	X	1. < nP, nP > 2. < nP, P > 3. < nP, P >
5. *Energy*	-	-	-	-	X

Table 6.6: Relevant Relations Employed in Considering Preferences in Table 6.5
(Entries show the relevant relationships taken from each of the previous rows' preferences as completed in Table 6.5)

Row	*Proximity*	*Image*	*Report_Size*	*Presence*	*Energy*
1. *Location*	X	{ n, nP, I, J }	{ n, nP, I, J }	{ n, nP, I, J }	{ n, nP, I, J }
2. *Image*	inv	X	{ n, nP, I, J }	{ n, nP, I, J }	{ n, nP, I, J }
3. *Report_Size*	inv	inv	X	{ nP, J }	{ n, nP, I, J }
4. *Presence*	inv	inv	inv	X	{ P, J }
5. *Energy*	inv	inv	inv	inv	X

Table 6.7: The Sets of Permissible Relations Arising from Intersection of Identified Relevant Relations shown in Table 6.6

The three tables are completed concurrently, resulting in an identified preference ordering of:

Presence *P* Image *P* Report_Size *P* Energy *P* Proximity

No inconsistencies arose during the completion of the tables, that is the decision maker did not feel that the available pool of permissible relations conflicted with the relation he wished to express. If it had, then he would have had to check over the statements made so far to see that they were, indeed, accurate. Equally, at the conclusion of the process, he felt that the ordering which had been identified was a fair interpretation of his beliefs in this respect. That is, in terms of the assumptions he made during the definition process, an indication of fauna presence would be the most valuable element in constructing an interesting, general-purpose display, together with an image. However, there is insufficient information to reveal precisely how the interaction between environmental conditions, sensor thresholds and, hence, camera activation will work exactly. That is to say, there might, on average, be only a single camera active, or they all might be continuously on, it remains to be seen, and there is no guarantee that their activation will automatically coincide with the emergence of the target fauna. Report_size is a fair indicator of node activity and will provide useful information to display graphically, but images and references to animal activity rate more highly in terms of visual display. Finally, the proximity to the assumed centre of activity, which was identified prior to any information being received about the area, is seen as being of least importance, less than the risk of a node failing and causing display inconsistencies, and, in addition, as it is a static attribute too much reliance on it alone may lead to lack of variation in the end display.

Having satisfied himself as to the appropriateness of this ordering, the decision maker can then continue to consider questions of evaluation coverage.

If the decision maker had felt it impossible to distinguish between the attributes, or had any feelings of insufficient strength to warrant such differentiation, then, for the purposes of subsequent modification, he would effectively have an open choice as to which attributes to drop, with the exception that, if an attribute were defined as being either mandatory or partially-mandatory, it must not receive such treatment. In this case, however, all the attributes were noted as being optional so the issue would not arise.

K - Define Coverage and Drift Rates: Establishing what level of coverage can be tolerated is an activity which is dependent upon the decision maker's wider knowledge about the system context within which the evaluation is to be employed, particularly with respect to how those factors which impact upon performance constraints are to be exercised. The more information available in this respect, then the greater can be the synchronicity between evaluation modification and other system elements. Identifying appropriate coverage rates decomposes into the tasks of identifying the sample size, considering what types of adaptive behaviour are worth supporting together with the flexibility of the evaluation over the sample period, and, finally, considering whether there need to be restrictions on the way that any dynamic modifications are applied to the evaluation.

Settling upon an appropriate sample size involves profiling the evaluation's intended usage, either on an informal or formal basis, so as to detect any common pattern which might make an appropriate sample candidate. A second factor to take into account is whether there is any cyclical pattern in the way the results of an evaluation are employed which might influence the choice of an evaluation cycle. Finally, if neither of these two situations produces an immediately obvious sample period, then other local but unrelated activities can be investigated to again see if any possible causes of intermittent disruptions to the evaluation emerge.

Having chosen an appropriate sample period, the next question is to decide what forms of adaptive behaviour to support and how to distribute their usage across the sample. The choice of the most suitable forms has already been identified as part of the analysis process. However, although more than one type of adaptive form may be cited as a possibility it does not necessarily follow that it is worthwhile, or indeed necessary, to accommodate all of them. Over-provision of adaptive behaviours is a waste of time and effort. The final choice of forms, and the way in which it is envisaged that their use can be tolerated across the sample period, will involve a consideration of the properties of the evaluation, such as its criticality, the number of attributes involved and frequency, and the way in which it is to be employed.

If it is felt that a certain number of full evaluations must be run during the course of the sample period, then it is reasonable to expect that their execution also follows a certain distribution in order to maintain a statistical balance across the evaluation's performance and for the full evaluations to exercise a steadying influence. Arriving at the shape of this distribution can be achieved by, again, giving consideration to the functional behaviour of the evaluation and the consuming process and will be tied in to the perceived integrity requirements of the evaluation process as a whole.

In this case, the evaluation objective is defined as non-critical and is, indeed, a straightforward and non-complex problem. It employs only five attributes and follows a highly cyclical pattern as it is synchronized with the summary report production rate. In the absence of any prior knowledge about both the behaviour of the external environment, which will be responsible for stimulating sensor activity, and about other network and system activity, then the assumption must be, at this point in time, that each cycle will encounter similar conditions. Therefore, there is no particular cause known in the external environment which could give rise to a need for differentiation in the application of the adaptive behaviours. Also, behaviour internal to the evaluation, that is the number of alternatives being presented each instance for assessment, is constant and, therefore, does not, in itself, generate any fluctuations in the workload either. The remaining influence on the setting of the sample size is that of the activity of the consuming process which, again, shows no differentiation in its usage of results across several evaluations. Finally, the decision maker decides to take into account how the display application might be used, particularly users' interaction. His informal opinion is that, as the information is to be provided at intervals of approximately 60 seconds, then this will reflect the display update rate, and most users will possibly watch the display for two or three minutes at the most unless something particularly interesting was being shown. Therefore, it would be advantageous to effect a change during this period, if for no other reason than to indicate that the system was still active. If the evaluation was to rely exclusively on only one or two attributes for an extended period of time, the probability of the results exhibiting a variability sufficient to effect such a change may be lower. This assumption may need to be proved. However, if the notions of coverage and drift rates are to be actively employed to monitor the evaluation then setting a very small sample size will incur marginally more overheads than a larger one, as counters have to be reset etc.. Taking all these matters into account, the decision maker decides to opt for using a sample period based on 12 instances of evaluation.

The decision maker then has to consider which of the possible forms of adaptive behaviours would be the most suitable to implement. The analysis so far has revealed that all four types of modification are feasible, that is skipping an evaluation entirely, dropping the number of attributes assessed, substituting an alternative method for the way in which an attribute is assessed and substituting an attribute itself. When the definitions of the five attributes are taken into account it is apparent that they are, already, in most cases the most direct ways of addressing the issues they explore. Therefore, to provide substitutions for the ways in which they are assessed, would be difficult to do in a more succinct manner. For example, a re-definition of the proximity attribute's current form of state appraisal might not be achieved without having to take into account several criteria rather than one. An alternative interpretation might be found for the presence attribute, but then the scheme, as implemented, would be faced with having to manage a single substitution variant along with the other adaptive types, which would add to the workload. A similar argument holds for actually substituting the types of attributes as well. The choice of attributes upon which to base the selection of data which the end display is to be built around, is that of the decision maker's. Other interpretations are, of course, available but, given the simplistic nature of this set, changes are unlikely to yield any significant positive performance differences. If there were more variation in the evaluation cycle and a greater level of complexity, then substitution of attribute type and/or its form of assessment, might be of more interest.

The remaining adaptive options are to either skip an evaluation entirely or drop attributes, forms of adaptation based upon the omission of actions and which are, consequently, more easily achieved. As none of the attributes are mandatory then either of these can be employed, but then, in the situation being addressed here, that would result in one of two things happening: either the display would not be updated, remaining static until the next set of data arrived; or, a default option would be run, which in this case is to be based upon node location. This latter action would give rise to the possibility of an interesting set of data being displayed, then an uninteresting set switched in as the default method was run owing to the evaluation being skipped on the following cycle, followed by a return to the interesting set when the evaluation was resumed at its next invocation. There is no significant fall out from this behaviour other than it may appear inconsistent to the end-user and, therefore, the decision maker applies his personal view that this should be avoided. This means that the skipping alternative is then also to be overlooked, leaving attribute truncation as the final option.

So, to summarise this consequence of his considering the problem so far, the decision maker has decided to set the sample period at 12 evaluation instances, to consider only attribute truncation and to set the upper bound on employing this approach as 2/3 instances. This leads to the observation that there is an apparent inconsistency here between the flexibility which an evaluation is, in theory, capable of showing and that level which it is finally allowed to exhibit. By stating that no instance of an evaluation can be skipped that can be effectively interpreted to mean that at least one attribute is mandatory, which is in conflict with the attribute definitions where they are all stated as being optional. However, there is a difference in meaning between the two states: the failure to honour a mandatory attribute results in the integrity of the result being undermined whereas no such implication is associated with the failure to observe the non-skipping requirement as imposed on optional attributes. In this case, the decision maker has stipulated that an instance of evaluation not be skipped, not because of his concern about the fundamental soundness and sufficiency of data handled in the evaluation, but because of the cosmetic effect of it not being run.

Having noted that the ideal, as far as he is concerned, ratio between degraded and full evaluations is 2/3 instances, the decision maker has then to consider which of the attributes he is prepared to allow to be dropped during the degraded portion together with how many of these can afford to be lost in any one instance. It should be remembered that this whole analysis process is being carried out before the actual evaluation methodology itself is chosen and the actual interrogation of the attributes undertaken by which the relative desirability of attributes is formalised. However, here preferential ordering is to be used to shape the attribute truncation process and so, by using the results of stage J which produced the preferential order, then the solution of which attribute to drop is partially resolved. Deciding how many to drop again requires the decision maker to exercise his individual judgement. After considering the problem description, and taking into account the small number of attributes employed, which means each might assume a greater responsibility in the production of a satisfactory solution than otherwise, he comes to the conclusion that an evaluation cannot afford not to assess the sate of the image, presence and report size attributes. Not only are these directly relevant to the subject of ongoing environmental monitoring, which is not the case for the remaining two attributes, but in addition the location attribute is static and is of the least relevance when dealing with the dynamism of the environment.

Once the structure of the coverage required of the evaluation is fixed, then it remains to apply the score by which to describe it in subsequent operations. The scores applied in order of inverse preference and using the same scoring scheme as discussed in Section 5.5.2 are as follows: location - 0.008, energy - 0.025, report_size - 0.074, image - 0.223, and presence - 0.669. From these, the minimum coverage score which can be expected, should the coverage rates be honoured, is shown in Table 6.8.

The values shown in Table 6.8 are the minimum levels of coverage which are deemed acceptable. Of course, expressing what constitutes a desirable level of treatment does not, in itself, do anything to guarantee that such treatment will be received. Both in the case of stating coverage/drift requirements and in stating that an attribute is mandatory, the enforcement of such statements is dependent upon the actual implementation of the evaluation process and its significance to the system. It remains for the decision maker to informally satisfy himself as to the appropriateness of his stated coverage and drift rates before proceeding to the conclusion of Section Two.

	Number of Instances	*% Sample*	*Sub-Period Score*	*Total Minimum Score Per Sub-Period*
Sample Size = 12	4	33%	1.00	4.00
	8	66%	0.967	7.74
Total	12			11.74
Coverage = 11.74 / 12 = 0.978				
Drift = 8 /4 = 2				

Table 6.8: Calculation of Coverage and Drift Rates for Sensor Example

L - Complete Problem Description: Before progressing to the actual selection of a suitable evaluation method, the description of the decision problem provided so far should be checked for completeness and revisions or additions made, where appropriate. The general nature of entries in the definitions so far can be exchanged for more detailed information as and when it becomes available during the course of the subsequent development process.

The salient points from the perspective of deciding upon a suitable form of evaluation methodology are summarised below. Awareness of these issues now will enable them to be accounted for directly, thus improving the efficacy of the evaluation as a whole.

- Evaluation type: reduction
- Comparative form: local
- Particular attribute characteristics of interest: potentially capable of being traded-off, there are no value dependencies, and they may be preferentially differentiated
- Ideal maximum amount of adaptability to be tolerated: coverage score = 0.978 {< 4, 1.0>, < 8, 0.967 > }, drift = 2
- Adaptability Form: Attribute truncation { Z1.proximity, Z1.energy}
- Evaluation rate - 1/60 seconds
- Problem size (maximum): five attributes, 10 alternative objects

6.2.3 Section Three - Part One: Summary of Method Characteristics

Section Three requires as input the definitions of the evaluation problem together with, ideally, a pool of potential candidates for the role of evaluation methodology. As was noted in Section 5.2.3, it may be that a presumption has already been made as to the type of evaluation methodology to be employed, in which case the focus of activities in Section Three becomes one of identifying what properties of the evaluation problem may not be supportable. Here, the assumption is that the three methods discussed are as in Chapter 4 and with the consequences of that examination forming the subject matter of the methods' characterisation.

Table 6.9 represents the responses to the set of example questions as proposed in Section 5.2.3, which are used to explore the suitability of the various evaluation methods on offer to the type of evaluation being considered. Depending upon the importance and scale of the problem, in a real situation the observations made in Table 6.9 might be backed up by evidence and cross-referenced to relative examples, here, however, such detail is absent. As, in this example, only a single decision maker is involved, the comments made reflect his interpretations and judgements alone. In scenarios involving multiple participants, some negotiation may be required before a definitive set of observations is arrived at.

***Evaluation Methods*:** **Weighted Additive Value Model (WAM) | Enumerated & Scoring Model (ENUM) | Lexicographic Based Approach (LEX)**	
Q1: How appropriate is this evaluation method to performing the type of selection required (reduction)?	
WAM	Suitable (1)
ENUM	Suitable - depends upon the granularity of problem interrogation at which the evaluation is defined. (1)
LEX	Unsuitable without repetition; there is a risk of the operation reducing the evaluation to the examination of a single criterion. (1)
Comments	(1) The ability to distinctly rank all alternatives will be in part influenced by value precision, range etc.
Q2: How amenable is this form of evaluation method to the named type of modification (attribute truncation)?	
a: Attribute Dropping	
WAM	Not directly amenable - theoretically would require a re-interrogation of the modified problem scenario to address new preferential state and consequent re-adjustment of weighting factors. (1)
ENUM	Amenable - but need to remember to account for amended total scores in expressions & consequent significance of values produced. (1)
LEX	Amenable - handling different number of attributes is an inherent part of normal operation. (1)
Comments	Although only attribute truncation is being considered here, the above comments would be equally applicable for the both cases of attribute substitution (type and form of appraisal). (1) It is important for the integrity of the specific evaluation instance affected that the modification is applied consistently across the alternatives being assessed.
Q3: What would the possible impacts of modification be on the usability, integrity and the validity of the results produced if modification in general were to be implemented directly with no extension to the method's definition to compensate for such change?	
WAM	Danger of misrepresenting value structure but still producing apparently viable results - difficult to detect inconsistencies as results are not functionally mapped to single alternative state; inappropriate substitution of attributes, or their form of measurement, without modifying value functions/weights accordingly would lead to misrepresentation of value structure; failure to generate satisfactory result in evaluations of global type owing to failure to adjust bounding conditions. (2)
ENUM	Failure to find satisfactory result owing to failure to account for changes in scores; inconsistency in applying modifications may lead to inconsistency in score; may give rise to grouping of results owing to coverage of fewer attributes which may not have been accounted for. (1) (2)
LEX	Inconsistent application of modifications; removal of inappropriate, i.e. most significant attributes; failure to note changes in basis for evaluation and not realise results represent different evaluation scenario. (1) (2)
Table 6.9: Querying Suitability of Potential Evaluation Methods Continued ...	

Q3 Continued:	
Comments	(1) If the problem is interrogated less extensively, i.e. by fewer attributes, then there may be less opportunity to distinguish between objects i.e. the results will exhibit bunching of objects as they are evaluated to the same level. If such behaviour is to be expected it will need to be accounted for. This is more of an issue for the ENUM and LEX approaches because of their tendency to be defined at a coarser granularity of evaluation. (2) Fluctuations in the number of attributes assessed may be expected to cause some fluctuations in the values assigned to alternatives between instances of evaluation. As a consequence, this may result in increased work for the consuming process if it results in a greater level of switching between alternatives after each evaluation instance.
Q4: How much work would be involved in investigating, specifying and implementing modifications and extensions off-line to the method's definition?	
WAM	i. The method has a more complex structure, directly including an investigation of evaluator's value system, therefore any alteration to the elements involved would, theoretically, require re-examination of the structure in its entirety for each modified problem scenario. ii. A conflict may arise between the preference order defined for coverage & attribute dropping purposes, if used, and that which emerges as a consequence of the weighted value structure as the two may not be identical. The alternative is to redefine coverage and order of attribute truncation to account for WAM interpretation of significance. (1)
ENUM	Little - need to revise scoring scheme to account for modified problem scenario. (1)
LEX	None if using pre-defined preferential ordering to order attributes. (1)
Comments	(1) Additional complexity associated with managing modifications.
Q5: What are the additional runtime overheads likely to be for the modified version over a non-modified version of the particular evaluation method?	
WAM	Runtime costs associated with dropping attributes would reflect cost of updating weighting constants and value functions. (1)
ENUM	Dropping attribute: identifying attribute to skip. (1)
LEX	Dropping attribute: identifying attribute to skip. (1)
Comments	(The precision of answers here requires knowledge of implementation; in the absence of this, general observations apply). (1) If active monitoring of coverage & drift are required then the number of operations necessary to manage the evaluation will be influenced by the dynamism of the environment.
Q6: How dynamically scalable is the function and what are the likely overheads to be given the known number of attributes and alternative objects stated in the objective definition?	
WAM	Highly scalable - can deal with any number of alternatives; demonstrated as requiring least number of operations of three methods; introducing additional attributes would require redefinition of value function and weights.
ENUM	Difficult to scale efficiently - no theoretical impediment to handling range of alternatives, only if detailed scoring scheme involved may be operationally slow; dynamic alterations to number of attributes easier than in WAM as can insert and remove expressions & adjust associated scores as appropriate, without disrupting ordering of other attributes; scalability depends upon granularity of expression which evaluation set at.
Table 6.9: Querying Suitability of Potential Evaluation Methods Continued ...	

Q6 Continued:	
LEX	Scalable - no theoretical impediment to handling increased number of alternatives, but was observed to be slowest of three methods in trials under the worst-case; inserting new attributes is feasible without disrupting ordering of attributes - in practice, however, may generate no observable difference in results if high order attributes do not generate ties between alternatives.
Comments	There are two different scenarios as regards the modification of evaluation on-the-fly; either the modification is foreseen, in which case, alternative value definitions can be provided to account for the modified problem scenarios and up-loaded accordingly - the cost of dynamic modification in this cases is largely involved with the overheads of identification of appropriate state and components to change; or, modification was not anticipated at the time of implementation/initialization and an alternative value structure has to be defined dynamically as well to cope with the revised problem definition. A response under this latter category might involve various logic/rules based approaches. The former case is the assumption here - that is the modification has been anticipated.
Q7: How inherently robust is the evaluation method, that is to say, what are the levels of integrity and usability associated with a set of partial results, as might be generated if the process is interrupted?	
WAM	Less robust; no indication as to what order the attributes are addressed in, i.e. least important first, most important or random so difficulty in identifying what individual elements have been investigated in a partial result.
ENUM	Reasonably robust assuming method implemented such that deals with attributes and alternatives in logical fashion - usability of a partial results, i.e. incomplete score, will depend upon whether it known what state the partial score represents; likewise, could make use of both completed alternatives and those which have only been partially assessed if again know what scores represent. (1)
LEX	Robust - at any point temporary result should only reflect identities of those alternatives which are still in the running. (1)
Comments	(1) In the cases of ENUM and LEX the usefulness of the partial results will be dependent upon whether the alternatives have been addressed in their entirety (depth first approach) or if each attribute is assessed across all alternatives in turn (breadth first). WAM more suited to breadth first approach so usefulness of partial results will depend on how many alternatives have been covered.
Q8: From observations, how would order methods on basis of efficiency?	
WAM	First (2)
ENUM	Second (2)(3)
LEX	Third (1) (2)(3)
Comments	(1) Capable of producing result quickly if single winner emerges as consequence of first attribute evaluated; would show relatively speedier performance, compared to other two cases, if evaluation problem defined a large number of attributes to be assessed. (2) Performance is influenced by problem size (number of attributes and alternatives) and on complexity of attribute appraisal methods. (3) Performance is also influenced by the actual attribute values encountered.

Table 6.9: Querying Suitability of Potential Evaluation Methods

In characterising the potential evaluation methods with reference to this specific problem, the decision maker develops a better understanding of how the two halves of the problem scenario, that is its definition and the manner of its resolution, are inter-related. Once this stage is completed, he can then proceed towards the conclusion of the analysis process with the selection of the most appropriate form of methodology.

6.2.4 Section Three - Part Two: Choice Procedure for Selecting Appropriate Methodology

Having considered the known capabilities of each of the individual evaluation methods being suggested against the identified required properties of the defined problems scenario, it remains to select one of them to actually undertake the job of evaluating the problem dynamically. In this situation the decision maker is also the person with responsibility for implementing the solution too, though this does not necessarily have to be the case. The decision maker's role is to provide the value structure which underpins the resolution of the problem and not, necessarily, to deal with the technicalities of building the final form of the evaluation.

The degree of formalism involved in arriving at the final choice will be dependent upon the development context, with such factors as number of decision makers involved, criticality of decision and business environment coming into play. The important point is that, having spent some considerable time on reviewing the various aspects of the problem scenario, there is a greater awareness of the nature of the problem and less likelihood of its integrity being undermined by the adoption of ill-considered approaches.

To illustrate the type of thought processes involved in an informal resolution of this question, the decision maker in this scenario considers certain characteristics of his problem and the pros and cons of following each of the evaluation methods in light of these. To start with, there are no constraints to take account of within the evaluation itself, and all attributes may tolerate being traded-off in favour of one of their peers. There are, however, temporal constraints in the environment to be dealt with. Scalability is not expected to be an issue, in terms of either the number of alternatives to be handled or with respect to attributes, as it is envisaged that the monitoring network will only be reconfigured as a consequence of failure or change of task. There are only a relatively few attributes, four of the five of which present continuously varying values whilst the fifth, proximity, which it is presumed will remain static for the duration of the network's deployment, and it is not known for certain whether their actual readings will present a sufficient variation in results to always generate a single winner. The presumption is that, given the precision at which values are assessed, this will be the case. However, it may be advantageous to consider approaches which can more readily explore this continuous nature. In addition, the initial objective, as defined, was to allow several issues to be considered in order to arrive at a solution, rather than focus on a single issue, and there is no reason to assume that extended reliance on methods which favour a single attribute resolution will be acceptable. Only a single form of modification is being proposed, that of attribute dropping, so choosing a mechanism which can readily support this approach would be a judicious move.

Given the previous points, each of the three evaluation methods has certain advantages and drawbacks. The WAM method is the most effective way to deal with the continuous distribution of values and also can succinctly represent the question of attribute trade-off. It does, however, involve a rather detailed definition procedure as a consequence of which both the decision maker's value structure and the numerical function are arrived at and, as the only form of modification to be employed is that of attribute truncation, there would be a necessity of dealing with the three variants of evaluation which might arise. To its benefit, though, this method has been observed as being the operationally quickest of the three and it works at a resolution which would facilitate a single alternative to be distinguished, should the underlying state of the alternatives permit it. Its other great advantage, namely its high scalability, is not required here.

The ENUM method does not have the complexity of definition burden associated with the previous method but equally may present problems if made to deal with a very fine granularity of value expression. For some problems this may not be required but, in this example, the exact distribution of attribute values, particularly the relationship between value levels and environmental activity, is not known and, therefore, it is difficult to identify how to score the various sets of values appropriately. Again, the scalability factor is not an issue though this method does not perform well in this respect. If the decision maker chose this method he would also have to provide the value model to accompany it, which might not be too much of a problem here as the problem is a simple one. The ENUM method is as capable as WAM in dealing with multiple attributes, it just depends on how detailed its definition is made which, as there is to be a single alternative selected, would need to be quite involved or address the key factors of performance in its scoring profile. The overheads associated with its operation would be better known once the level of investigation was established.

In contrast the LEX method is not best suited if a thorough exploration of multiple attributes is required each instance of evaluation. Indeed, depending upon how the attributes' values change, it may seldom consider some attributes at all. It does, however, suit the production of a single winner where attribute states permit it. Although scalability is not of concern here, this method has been shown to be the slowest of the three available options, which would place it at a disadvantage in terms of the known environmental constraint. It also cannot represent trade-offs between values as would ideally be required of the method selected here if the full flexibility of the evaluation is to be exploited. Its strengths, though, should not be overlooked, particularly its straightforward operation which could easily accommodate the preference ordering, already identified by the decision maker, as the way to deal with the attributes.

On the basis of the above points, the decision maker opts for the WAM method, specifically for its appropriateness with respect to allowing trade-offs between attributes and its ability to produce a single winner, especially as in this case he is not sure what are the critical value points which are to be encountered for. He also considers it to be the quickest of the three methods. It does mean, as a consequence of this choice, that he will have to handle the definition process with more thoroughness than might have been required of either of the other two methods but he is tackling the problem alone and, therefore, does not have to worry about trying to take into account several different opinions.

6.3 Comments

In terms of the design process, the activities associated with the analysis framework might be positioned at the unit design level, as shown in Figure 6.1, that is they relate to the specification and decomposition of the intended evaluation task. How such activities are approached in a real design scenario is seldom covered in the literature, and, as has already been mentioned, the thought processes which lie behind the structuring of value production are not necessarily fully described either. If one takes by way of comparison the example discussed by Clarke et al [Cla99] when they were deriving the value structure for the Airborne Tracking System, one might make certain observations. In the cited work the focus was on the identification of the value structure and then upon the production of value functions themselves, rather than in this case where greater emphasis is placed upon exploring the characteristics of the problem and the consideration of an appropriate method by which to identify values. Here, the actual identification of values is a consequence of a suitable method having been selected. Hence, there is less evidence in the former work of the reasoning which underpins the evaluation. It is assumed, from the description, that the interrogation of the decision makers was performed by people with experience in modelling the decision process, but even so, proceedings appeared to progress in a somewhat ad hoc fashion. Therefore, attempting to handle both the definition of the evaluation process and concurrently deal with the question of its adaptation may have been excessively confusing to the decision makers. Dealing with the design of either aspect may be problematic and, consequently, dealing with them collectively is best done in a structured manner.

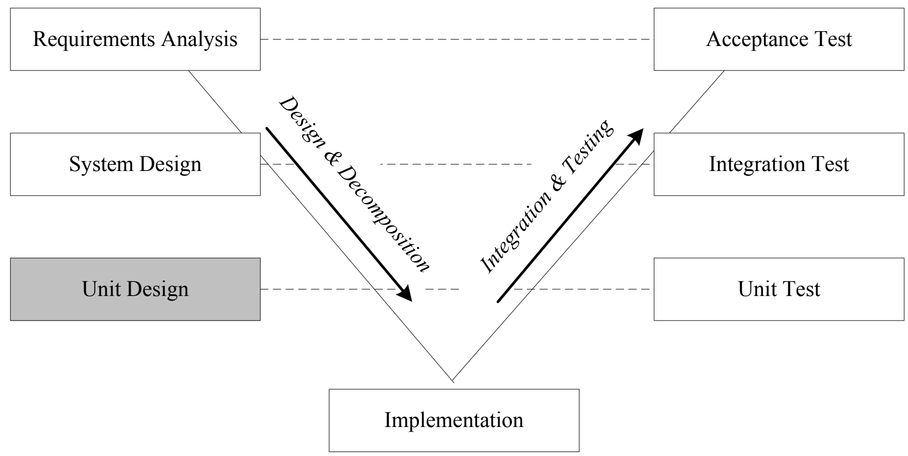


Figure 6.1: V-Model of Process Design Showing Level of Activity Appropriate to Framework

6.4 Summary

It is proposed that by analysing the proposed evaluation problem, as has been demonstrated, certain advantages are accrued to the development process. Firstly, there is a raised awareness of the need to maintain the integrity of the underlying value structure of the decision maker when dealing with the question of the problem's form of implementation. Secondly, it enables factors which might constrain a performance to be accounted for from the outset of the definition process such that the ensuing realisation of the problem is sufficient for its purposes rather than either being over described at one extreme, or inadequate at the other. Finally, the third advantage is that it allows any flexibility within the problem definition to be exploited as an integral part of the definition process and in a way that it does not reduce the integrity of the evaluation, rather than such flexibility either being ignored, thereby reducing the robustness of the evaluation, or ad hoc attempts to retrospectively deal with the issue which may give rise to inefficiencies.

This section has provided a worked example of the analysis methodology introduced in Chapter Five, and provided some insight into how to approach the problem of fixing upon the appropriate level within a system context at which to set the evaluation process.

Chapter 7

Integrating Adaptive Capabilities into Problem Evaluation

Having considered the theoretical aspects of undertaking a dynamic multi-attribute evaluation it remains to explore the practical implications of such activities. The application of hitherto untested theory is required in order to demonstrate the supposed benefits which are being claimed and also to reveal any inconsistencies and oversights in logic. This section undertakes an empirical exploration of the adaptive aspects of a multi-attribute evaluation, as discussed in Chapter 5, with the intention of showing that adaptation can be of benefit both to the integrity and predictability of the process.

One by-product in enabling a process to accommodate different performance levels is that it may permit this new-found flexibility to be exploited in different ways. Such exploitation can be interpreted as representing different strategies of behaviour where each strategy reflects the evaluating process' attitude towards observing the integrity of an evaluation. To this end, four strategies have been identified which describe different approaches towards the management of evaluation.

Indifferent: Indifferent evaluations ignore the potential for disruption to the evaluation and make no allowances for such events when planning the way in which alternatives and attributes are approached. That is, they arbitrarily decide whether a depth or breadth-first approach is to be assumed and what order attributes are to be treated in, with the assumption that, once settled upon, the approach is fixed for the duration of the evaluating process' existence. This situation basically represents that of embarking upon the definition and realisation of a dynamic evaluation scenario in an uninformed and unaware manner and, as such, really indicates a lack of a positive strategy.

Passive: A marginal improvement on the previous case, a passive strategy is one in which the evaluation problem is approached with off-line awareness, that is during the design and specification phase, but the on-line components have no runtime management capabilities. Thus, they neither monitor nor respond to fluctuations in service levels, making their performance dependent upon the manner of their definition. However, as a consequence of the increased awareness of the issues relating to the description of such decision problems, the definition itself can take a more reasoned and justified form. The intention behind adopting a passive approach is to endeavour to improve runtime behaviour without accruing any additional runtime overheads. The primary way in which this might be achieved is through consideration of the attribute definitions and ordering of attributes for evaluation purposes.

Aware: In contrast to passive and indifferent strategies, which are not dynamically adaptive and employ no notion of coverage and drift measures, an aware strategy incorporates both these aspects. It employs adaptive alternatives, e.g. dropping and substitution of attributes, and has an awareness of the theoretical coverage and drift requirements which allow it to monitor the performance of the evaluation instances. When confronted with a reduction in its computational requirements it can respond by selecting an appropriate level of behaviour from the range which is defined. It does not, however, indulge in co-operative interaction with other tasks to ensure that its minimum requirements are met and, therefore, cannot prevent total failures occurring. It also, does not actively manage the coverage and drift requirements so performance in this area is entirely dependent upon the state of the underlying attributes.

Active: Adopting an active adaptive strategy requires not only that the integrity boundaries of the evaluation be established and the appropriate adaptive form identified, but it also implies that on-line monitoring and management of the evaluation is undertaken. Thus, an active scenario would see an evaluation keeping a check on its on-going performance via the application of the coverage values so as to identify the appropriate form of modification to adopt in any specific instance. It would also co-operatively interact with the system to ensure that the minimum mandatory requirements were maintained reflecting the evaluation's perceived importance. Active management of the adaptive process, on a co-operative basis, enables both coverage and drift levels to be observed and thus ensures that each instance of evaluation should not breach the integrity of the evaluation. The consequence of this improved responsiveness is that, unlike the passive and aware strategies, there will be a greater management overhead to account for.

These four strategies are employed in the following tests so as to enable any incremental advantages in undertaking the reasoned implementation of a multi-attribute decision problem to be seen. It cannot be expected that all such problems will either necessitate or tolerate the employment of an *active* strategy but it may prove to be advantageous to consider one of the lesser options.

Whilst the focus has been on employing an adaptive evaluation in order that it can respond to a known constraint in its runtime environment, it may also prove beneficial to enable such approaches to deal with occasions where the evaluator fails to access particular elements of the evaluation, such as when a particular attribute or alternative is temporarily unavailable. Such events are more likely to occur if the evaluation involves distributed objects where components may be experiencing communication and location difficulties. Being able to make use of the group of elements which is available in these situations to ensure a viable result is generated, albeit of a reduced quality, would enable performance to be maintained.

The primary objective motivating the following examination is to see whether even undertaking the minimum of additional effort can achieve an improvement in the robustness and behaviour of the evaluation over that seen when the structuring of the problem is approached in an entirely arbitrary fashion.

7.1 Analysis

The model employed to explore the application of the four strategies, discussed earlier, was as follows. Two tasks were defined each with a period of 100 units and with their deadlines equal to the period length. They were scheduled under a fixed-priority scheme, with Task 1 having a priority of 1 whilst Task 2 had a lower priority of 2, meaning that Task 1's requirements were predominant. Task 2 represented the task performing the evaluation and, therefore, had to deal with the fluctuations in the service it received, wherever possible. These fluctuations were generated by varying the computational requirement of Task 1 which effectively acted as a placeholder for all other computation not directly involved in the evaluation and was, therefore, in competition with Task 2 for access to resources. The use of two tasks can be justified on the grounds that the intention was to focus upon the adaptive capabilities of Task 2 rather than inter-process communication protocols. Task 2 is only concerned with the size of its resource allocation, not the reasons why this quantity should fluctuate which are beyond its scope.

Attribute ID	*Mandatory Level*	*Full Variant Constraint Value*	*Full Score*	*Partial Variant Constraint Value*	*Partial Score*
Att_1	Mandatory	4	0.66677	0	0
Att_2	Optional	4	0.00091	3	0.0003
Att_3	Optional	4	0.00823	3	0.00274
Att_4	Partial-Mandatory	5	0.07409	3	0.0247
Att_5	Mandatory	3	0.22226	0	0

Table 7.1: Attribute Characteristics (values relate to evaluation across all alternatives)

Five attributes were defined, three of which (*Att_2, Att_3* and *Att_4*) also had substitution variants described by which the evaluation might exhibit some flexibility in its operations. The attribute characteristics are shown in Table 7.1 from which it can be seen that three mandatory attributes were specified. Twenty alternative objects were assumed. The attributes were preferentially ordered *Att_1* P *Att_5* P *Att_4* P *Att_3* P *Att_2*. When the combinations of full and substitute attributes were taken into account a possible six different levels of performance emerged, as detailed in Table 7.2. It is worth noting that, where an attribute has both a full and partial form, whilst the preferential ordering is done on the basis of considering the attribute in its entirety, at runtime the full variant is dropped before the partial variant. Therefore, for the purposes of adaptation, the variants take priority over the full attribute interpretations, as can be seen in level vi in Table 7.2. It should also be remembered that, the objective in applying preferential ordering in the first place, is to enable the structured management of adaptation via the application of such an ordering. Consequently, those permutations which do not conform with the ordering and undermine the problem definition are removed from consideration, hence the omission, for example, of the combination < *Att_4, Att_3, Att_2* > from Table 7.2 which omits to address the mandatory attributes.

Adaptive Levels	*Cumulative Score*	*Attributes Employed*							
		Att_1	*Att_5*	*Att_4s*	*Att_4*	*Att_3s*	*Att_3*	*Att_2s*	*Att_2*
i.	0.9723	X	X		X		X		X
ii.	0.9717	X	X		X		X	X	
iii.	0.9714	X	X		X		X		
iv.	0.9659	X	X		X	X			
v.	0.9631	X	X		X				
vi.	0.9137	X	X	X					

Table 7.2: Evaluation's Adaptive Levels following Preferential Ordering (s = substitute variant)

Coverage Boundaries	*Number of Instants*	*Coverage Score*	*Attributes Employed*
Level 1	3	0.9723	Att_1, Att_2, Att_3, Att_4, Att_5
Level 2	4	0.9714	Att_1, Att_3, Att_4, Att_5
Level 3	3	0.9137	Att_1, Att_5, Att_4s
Drift Value = 3			
Table 7.3: Coverage and Drift Values			

The attributes were scored using the schemes described in Section 5.5 and 5.6 to produce the coverage and drift values, the details of which are given in Table 7.3. Three of the scoring levels from Table 7.2 were identified to represent the tolerated ranges of the evaluation's performance. That is, the maximum coverage provided when all attributes were run in full achieved a score of 0.9723 whilst, at the other extreme, the minimum level of coverage which could be tolerated was represented by running only *Att_1, Att_2* and the partial variant of *Att_4*, resulting in a score of 0.9137. The minimum case represented coverage of the mandatory set of attributes alone. The scoring levels specified in Table 7.3 represent the lower bound on evaluation performance for the stated number of evaluation instances. Therefore, those performance levels listed in Table 7.2 which do not map directly to a scoring level specified in Table 7.3 are still allowable if they occur in the appropriate interval. The sample size was set at 10 and the evaluation was assumed to be run once only at the point of Task 2's arrival. The coverage score associated with each attribute assumes that it has been appraised across all alternatives. In order to assess the score, during an evaluation instance each attribute score must be divided by the number of alternatives being used so as to obtain the per alternative element.

Two different workloads were employed in the tests, a static version which had been generated randomly but then fixed so as to allow comparison between different strategies, and a randomly generated workload. The static variant equated to 100 instants of evaluation, making 10 sample periods, whilst the random variant ran for 1 million time units thus allowing 10,000 evaluations and 1,000 samples to be taken. Unless otherwise stated, the computational requirement of Task 1 was set at 85 units, and the static and randomly generated component, which was added to this in order to generate the fluctuations in workload, constrained to be between 0 and 10 units. This meant that in no situations could Task 2, with a maximum computation of 20, complete a full evaluation and in many cases it could not even maintain the mandatory level without adapting.

The four different strategies were run under both depth and breadth approaches. The four strategies are identified as follows: Indifference - *S1*; Passive - *S2*; Aware - *S3*; and Active - *S4*. For *S1* a random ordering of the five attributes was identified as *Att_4, Att_2, Att_5, Att_3* and *Att_1*. For the remaining three strategies, the attributes were handled under the preferential ordering.

Strategies 1 & 2

For *S1* and *S2*, the notion of coverage and drift has no meaning. However, the performance was scored as if it did in order to assess to what degree the results of evaluation would cover the issues of importance when resolving the problem being evaluated, remembering that the attributes which are of more importance to the identification of a satisfactory solution have a higher score. Charts 7.1 and 7.3 show the proportion of attributes assessed and the associated hypothetical coverage score achieved for the static and random cases respectively, using either the breadth or depth approach. For the latter, where each alternative is evaluated in full, there is no difference between the strategies as the time taken to process a single alternative is the same in each case. For the breadth-first approach, however, there is a difference owing to the strategies assessing the attributes in different orders. The time taken to evaluate an attribute is a natural consequence of its definition. However, in the breadth approach, should the favoured (that is high preferentially ordered) attributes also be those which are the quickest to measure, then the evaluation will be able to deal with more attributes. The difference between the two strategies is revealed under the breadth approach when considering their respective coverage scores. These show that, for no additional on-line effort, *S2* can result in more of the favoured attributes being evaluated. The number of times individual attributes are assessed is shown for each workload in Charts 7.2 and 7.4. For the depth-first approach there is no difference for *S1* and *S2* under the static workload whilst for the random workload there is a marginal difference of 416 elements in *S2*'s favour, though this is not revealed on the graph. For the breadth-first approaches the frequency at which each attribute is used is dependent upon its position in the order, with both strategies failing through lack of time to investigate the last attribute in their respective lists. It can be concluded from these observations that there is no difference between *S1* and *S2* if employing a depth approach and that this tactic does allow for consistent treatment of alternatives. However, only 8 alternatives were actually covered in their entirety this way under the static workload, which is 40% of the total number available. At least, under the breadth approach, both strategies obtained some information on all of the alternatives, each covering one attribute in its entirety.

Strategies 3 & 4

Strategies *S3* and *S4* are enabled with a range of responses which will allow them to accommodate a variety of service levels. Charts 7.5 and 7.6 again show the proportion of attributes and coverage scores achieved for the static and random workloads respectively. In

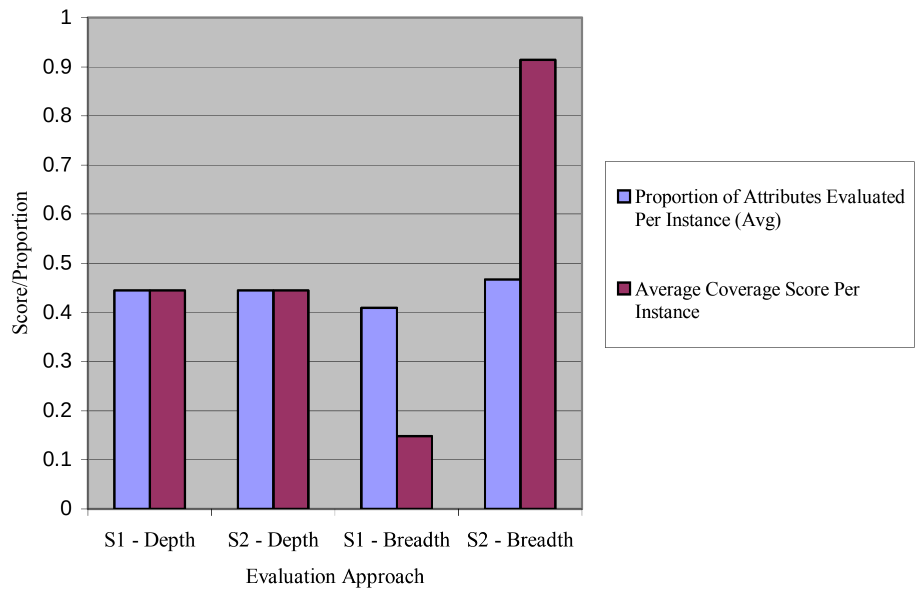


Chart 7.1: The Proportion of Attributes used per Evaluation against Hypothetical Coverage Score under each Strategy - Static Case Strategies 1 & 2

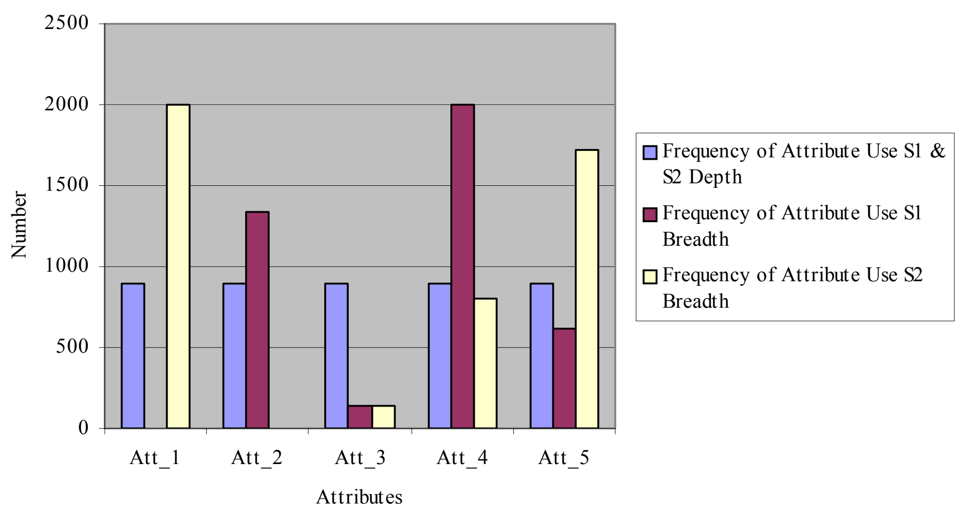


Chart 7.2: Frequency of Use of each Attribute under Different Strategies - Static Case Strategies 1 & 2

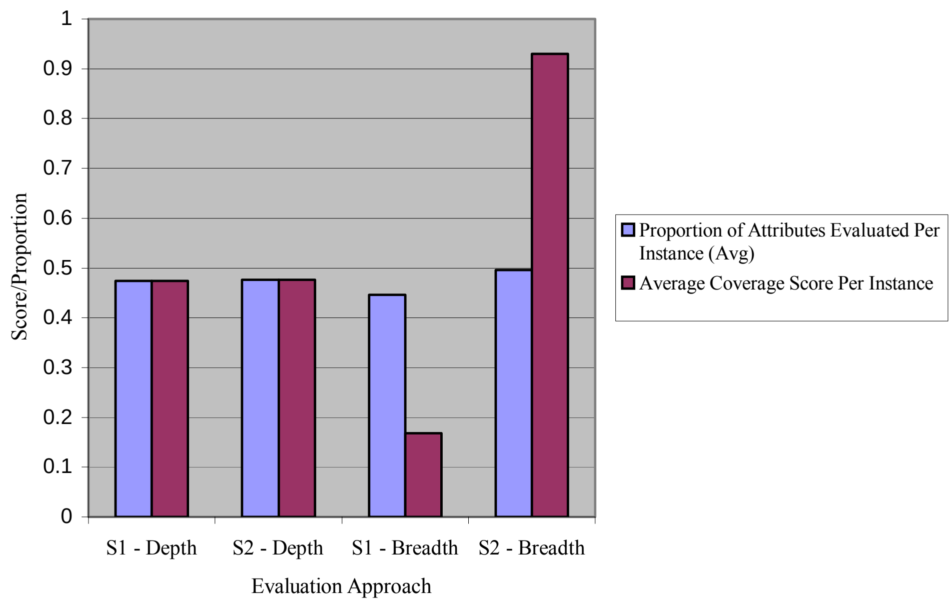


Chart 7.3: The Proportion of Attributes used per Evaluation Against Hypothetical Coverage Score under each Strategy - Random Case Strategies 1 & 2

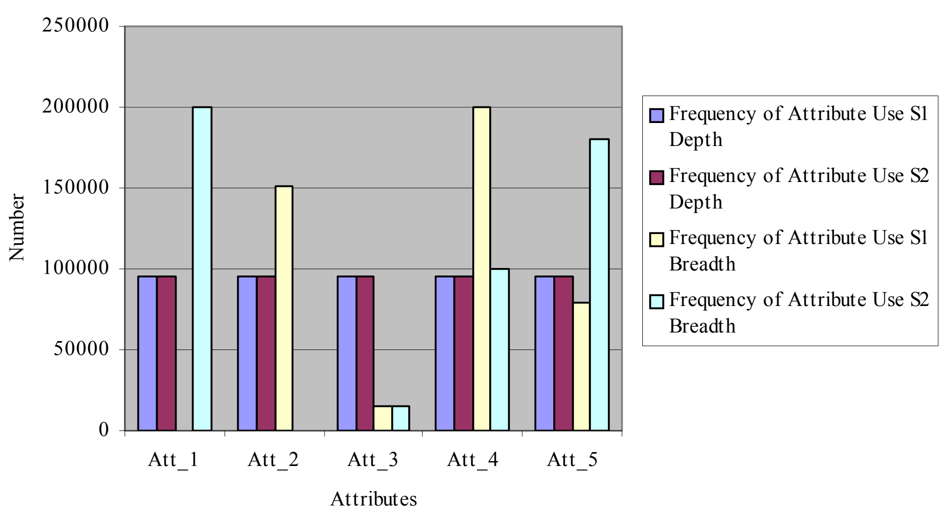


Chart 7.4: The Frequency of Use of each Attribute under Different Strategies - Random Case Strategies 1 & 2

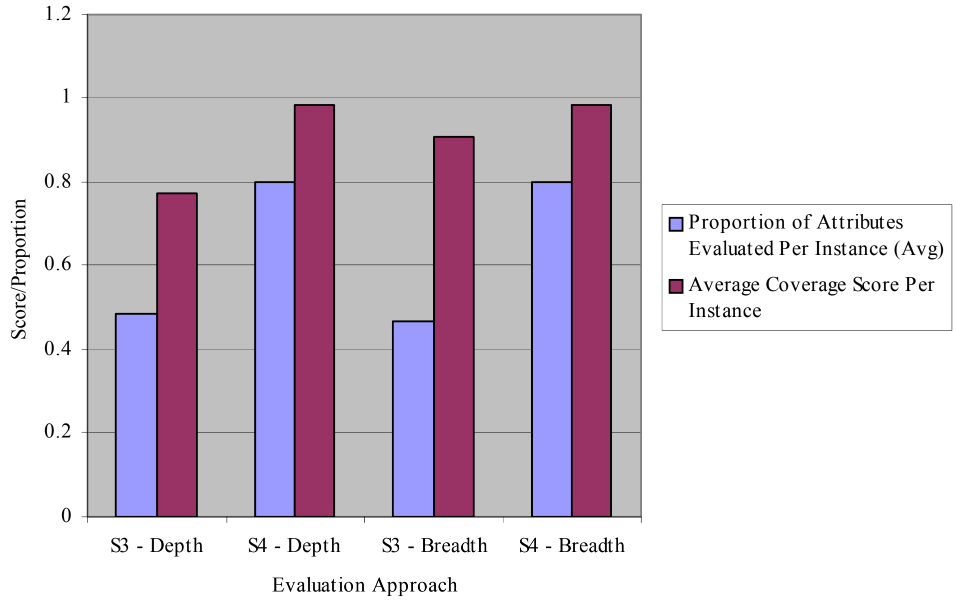


Chart 7.5: The Proportion of Attributes used per Evaluation against Coverage Score under each Strategy - Static Case Strategies 3 & 4

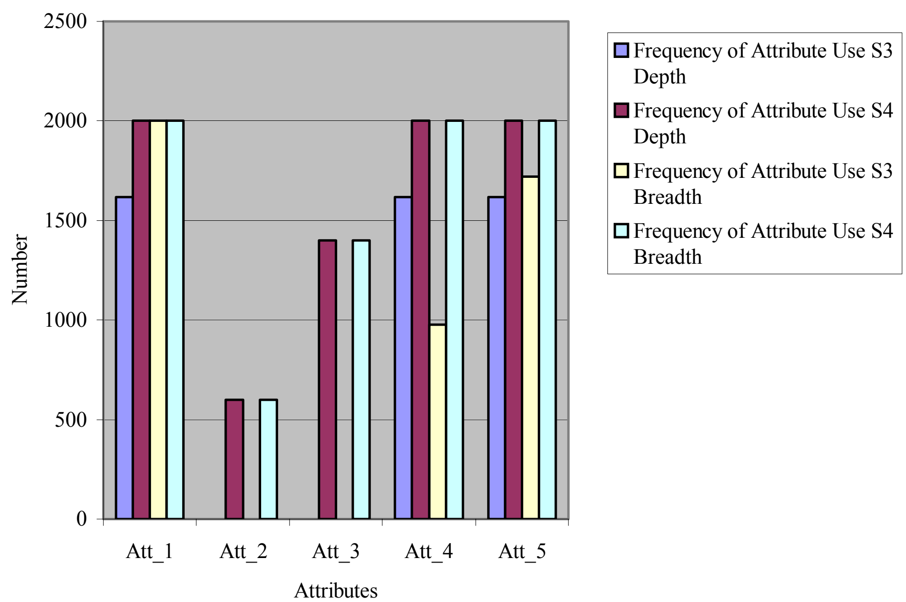


Chart 7.6: Frequency of Use of each Attribute under Different Strategies - Static Case Strategies 3 & 4

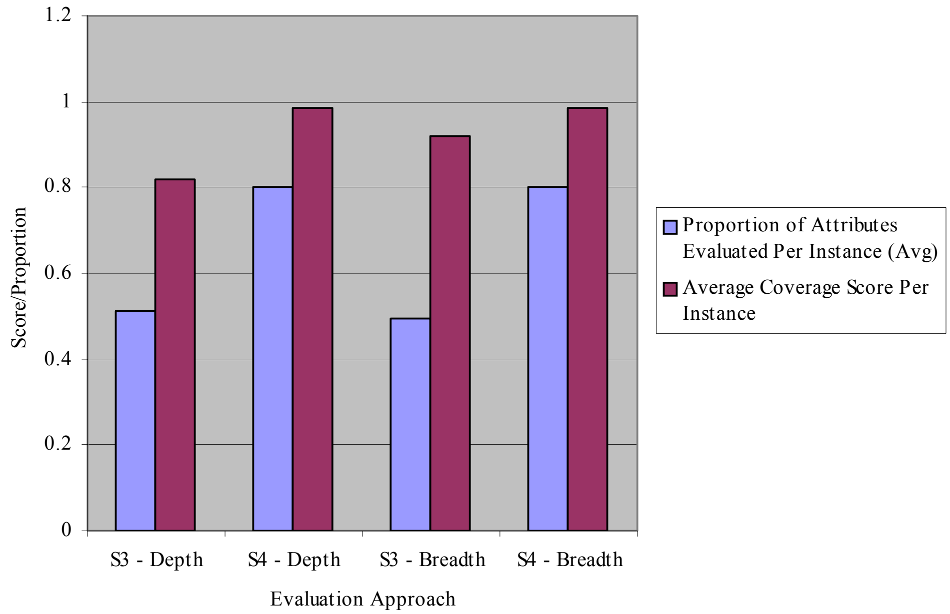


Chart 7.7: The Proportion of Attributes used per Evaluation against Coverage Score under Each Strategy - Random Case Strategies 3 & 4

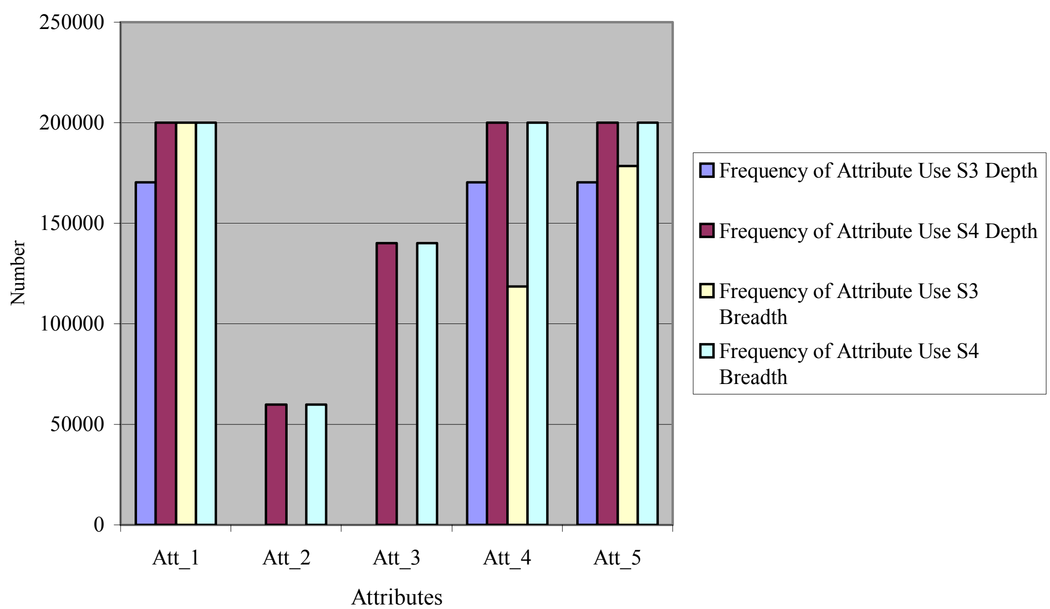


Chart 7.8: Frequency of Use of each Attribute under Different Strategies - Random Case Strategies 3 & 4

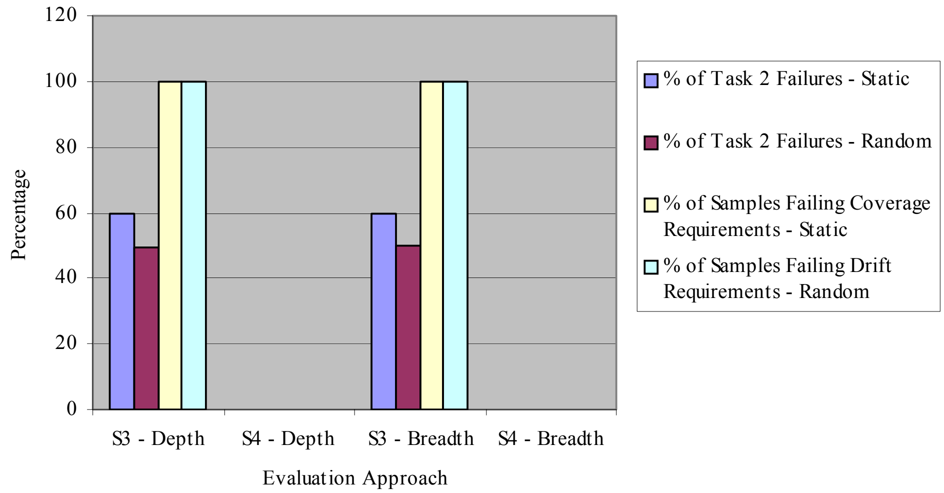


Chart 7.9: Task and Coverage Performance - Random Case Strategies 3 & 4

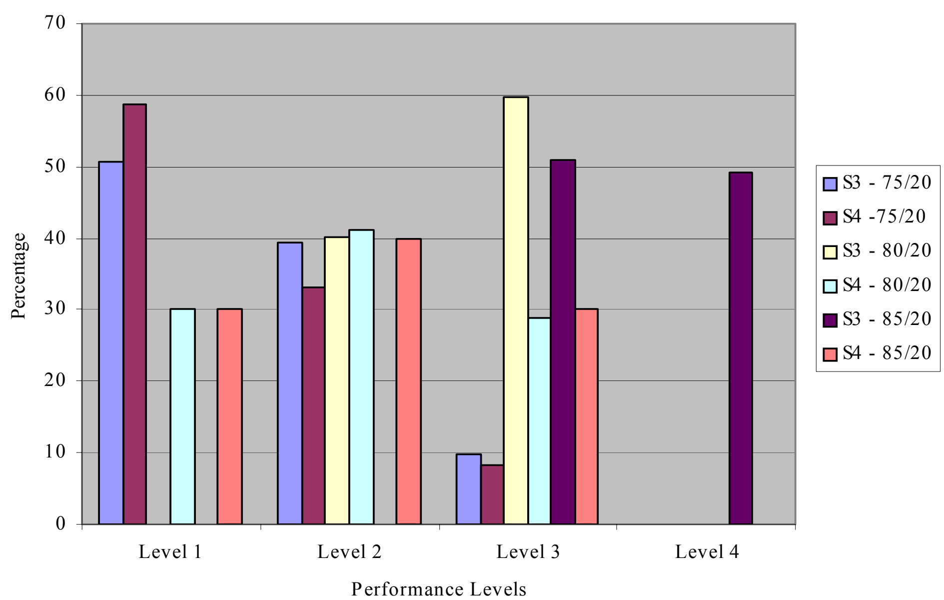


Chart 7.10: Frequency of Evaluation Instances at Different Scoring Levels for Different Computational Values of Tasks 1 & 2

Level 1 = max score; Level 2 = mid ≤ score < max score; Level 3 = low ≤ score < mid; Level 4 = < low score range this case *S4* assesses more attributes owing to the fact that it is able to maintain the time available to it at the minimum level required to allow it to handle all alternatives. Whilst *S3* can select a level of evaluation to perform at within the adaptive band ranging between the Levels 1 and 3 it cannot prevent the time available to it falling below that required for Level 3 and, therefore, there are occasions where it is unable to deal with all the alternative objects. Because *S3* and *S4* are able to employ less costly variants, in terms of the constraint, for the partial and optional attributes, they are able to more frequently access all of the attributes, including that at the tail-end of the order, *Att_2*. Consequently, in contrast to *S1* and *S2*, *S3* and *S4* are able to achieve higher coverage scores because they are able to employ a more economical approach when required and target the most significant attributes.

The real benefit, however, in terms of managing the coverage of an evaluation process is that it enables performance to be more predictable. Table 7.9 shows how the strategies meet coverage and drift requirements. *S3* has no ability to police this feature, only recording when breaches occur which can be for any single evaluation within a sample period. *S4*, however, controls this situation by monitoring current levels and setting the next minimum level required according to the coverage statement. If the requirement were for Task 2 to at least assess all alternatives to the minimum level then it would experience approximately 60% and 50% failure rates for the static and random cases respectively whereas under *S4* these targets would always be achieved. Also, it can be seen that as *S4* is capable of actively managing the evaluation then the question of whether to pursue a depth or breath approach becomes less significant as the coverage specification will result in the same treatment of the problem.

Of course success, particularly for *S3*, is governed by the actual nature of the constraints which, in this case, are dependent upon the amount of work required by Task 1. Chart 7.10 illustrates how the frequency at which the evaluation is performed at any given level naturally fluctuates with the given constraint. This chart considers three scenarios where Task 1's computational requirement is at least 75, 80 and 85 units. When the random element is added, this influences the amount of spare time available to Task 2 accordingly, and the chart shows the frequency with which evaluation falls within the different performance bands. Level 1 relates to the full performance being achievable, Level 2 the range between the mid and upper scoring levels, Level 3 the range between the lower and mid scoring levels and Level 4 the case where there is insufficient time to achieve the lowest scored performance. *S3* only falls within the Level 4 band when conditions are particularly stressed, the remaining time it can modify its evaluation to meet the current constraint level, showing that even without co-operative interaction and active policing there is a benefit to evaluation.

7.2 Observations

The advantage in using a scoring mechanism over appraisals based upon ascertaining the percentage of elements covered is that they contain more information about the nature of the elements included in the evaluation. Of course, if there is no notion of differentiation between elements then scoring is of no advantage. As has been noted, the scoring scheme described takes an attribute based approach with the score assigned to an attribute being predicated on its being appraised on all attributes of that type across all alternatives. This is only one of many alternatives which might be possible. Another interpretation might still employ the scoring scheme as described only, instead of thinking in terms of specific attributes covered think rather in terms of the number of alternatives covered. Thus, a full coverage (equivalent to Level 1 in Table 7.3) would necessitate evaluating all available alternatives, whilst the lower levels would allow for a fewer number being investigated. Additionally, on top of this might be applied different tolerances for the number and type of attributes covered as demonstrated in Section 7.2. This leads to the possibility of constructing a hierarchy of possible responses, thereby increasing the adaptive capacity of the evaluation and enabling further exploitation of the constrained level of services it receives.

Following from the previous point it can be seen that there are a further two forms of adaptive behaviour which have not been considered for active dynamic application. These are:

- Reducing the number of alternatives appraised in any single evaluation instant.
- Deciding whether to follow a depth or breadth approach.

Reducing the number of alternatives was previously deliberately excluded from consideration on the grounds that there was no justified basis for performing this action. However, this may be too extreme a demand as, in reality, the decline in service might effectively bring about a reduction anyway. The question of which approach to take, depth or breadth, was hitherto assumed to be an off-line, design issue. However, it may be advantageous to support the on-line exercising of this choice. For example, if the evaluating process is subject to a time constraint and must consider a mix of alternatives, some locally and other remotely based, then the discrepancy between processing time and communication time might favour treating the local alternatives in depth rather than attempt to assess all alternative objects via the breadth-first approach.

7.3 Summary

This chapter has applied the adaptive mechanisms described in Chapter 5 to a task based simulation with the objective of investigating their feasibility. It has demonstrated that by employing a reasoned approach, and without any additional on-line support, an improvement in the ability to investigate the decision problem can be achieved. By actively managing the investigation process, the evaluating task can maintain its own performance within acceptable bounds should other tasks be enabled to co-operate with it. By employing the notions of coverage and drift it is possible to allow the decision resolution to be dynamically adaptive whilst, at the same time ensuring the evaluation process behaves predictably. By having a deeper awareness of the nature of the evaluation problem and of the range of adaptive mechanisms which can be drawn upon then it is possible at the design stage to select a level of adaptive support appropriate to the importance of the evaluation process rather than waste resources by over-provision. A consequence of a lack of knowledge of these issues is that, for the want of some careful thought about the structuring of the evaluation process, performance may prove to be less acceptable than it otherwise might have been.

Chapter 8

Conclusion

The next generation of systems will be required to exhibit greater levels of self-management and to undertake the automatic resolution of problems encountered on-line. Such behaviours will necessitate that, amongst other skills, systems must have decision resolution capabilities together with the means to effectively manage the process by which such decision problems are tackled. This thesis has considered some of the issues relating to the employment of multi-attribute decision scenarios in the context of a real-time system and has proposed mechanisms by which their realisation can be structured in a methodical fashion. This chapter presents an overview of the work carried out and suggests related topics which merit further attention.

8.1 Summary of Contributions

To paraphrase the thesis proposition, as contained in Chapter 1, if multi-attribute decision problems are to be resolved effectively, and to the satisfaction of the decision maker, then their definition needs to be consistent in the treatment of the decision maker's preference structure which forms the basis for the identification of a satisfactory solution in the first instance. This can be said of any decision problem, whether it arises in a static or dynamic context. However, there will be additional complications associated with the dynamic execution of decision resolution methods which also need to be taken into account. For example, there are operational limitations which arise as a consequence of working on shared resources and which cannot be allowed to repeatedly undermine the integrity of the evaluation process. An acceptable response by which to enable any process to cope with fluctuations in its runtime environment is to allow it to adapt to the circumstances it encounters as, and when, they arise. However, retro-fitting adaptation to a decision/evaluation process runs the risk of compromising the integrity of the underlying preference structure and any properties which might be required of the evaluation methodology implemented, not least because of the relations between the multiplicity of issues involved. It is more effective, therefore, to deal with both aspects simultaneously, that is to ensure that the implementation of the process responsible for the evaluation of the decision problem is sufficient for the type of problem it encounters whilst, at the same time, using the knowledge gained during the definition phase, specifically with respect to the tolerances of the problem definition, to exploit any adaptive capacity which is present. In order to facilitate the realisation of such an integrated objective this work has proposed the following solutions:

A Framework to Guide the Definition Process: A methodology by which to guide the investigation of the decision problem in the first instance is described in Chapter 5. This has the particular objectives of ensuring that both the requirements of the decision maker are understood and the final form in which the evaluation process is realised proves adequate for the purposes intended. A secondary objective is to provide guidance to those people who have been given the task of resolving the definition and specification of the decision problem. Given the frequency and ubiquitous nature of decision problems, it is unlikely that this task will always be allocated to those with specialist knowledge of decision resolution techniques. Indeed, it is more probable that it will be devolved to available designers and engineers on an ad hoc basis. However, without solid models to follow, and models which are also realistic in terms of the demands they make on their adherents, there is the potential for inconsistencies to enter into the definition process. By employing a structured method to deal with the problem not only is the probability of such inconsistencies reduced but the level of awareness of those involved in the design process will be increased. Of course, the immediate effect of imposing guidance methods on any operative is to actually increase their workload in the short term, rather than reduce it. This disadvantage has to be weighed, though, against the cost of having to subsequently revise the implemented process when it proves inconsistent in its operation and fails to perform satisfactorily. Also, once familiarity with design methodologies and other operational tools is established, then their adoption reduces the level of uncertainty attached to any activity. Chapter 6 provided an illustration of the application of the framework. One element of the proposed framework is the provision of knowledge relating to the characteristics of the possible evaluation methodologies. A demonstration of how this knowledge might be acquired was given in Chapter 4.

Identified the Adaptive Possibilities of Dynamic Decision Problems: The argument for employing structured methods is that they improve the results of operations, partly by reducing inconsistencies. Kontio pursued just such an argument when considering component selection [Kon95a]. There is a second advantage, however, in undertaking a more rigorous exploration of the problem rather than attacking it in an unprincipled manner. That is, as demonstrated here, it allows certain traits, namely its adaptive flexibility, to be exploited to the advantage of the evaluating process. This work has illustrated some of the possibilities by which the evaluation of a dynamic decision problem might accommodate adaptive behaviours whilst retaining a sufficient level of service to still be of benefit to the system/decision maker.

A Set of Mechanisms to Facilitate the Management of Adaptation: There are two aspects to enabling adaptation in any process. One is the off-line specification of the adaptive range and the other is the on-line management of adaptation. Chapter 5 presented several mechanisms by which adaptive behaviours could be both brought about and then managed dynamically. The benefits of making an evaluation process adaptable were demonstrated in Chapter 7.

Perhaps the most beneficial contribution of this work, however, is to raise awareness of the subject matter, and to possibly stimulate further research in this area. For example, Chapter 2 mentioned some criticisms of common evaluation techniques which continue to be used, despite their inappropriateness, largely owing to the absence of more amenable alternatives. The existing literature either acknowledges the existence of time constraints on decision making but assumes a decision theoretic stance without considering the nature of the problem in a holistic fashion, nor the practicalities of managing multiple attributes [Bod94][Hor87], or it recognises the benefits of employing dynamic decision evaluation processes, but arbitrarily adopts evaluation methodologies with no evidence of having thought out the implications or suitability of their choices [Cha02]. No evidence was found of structured attribute management as a means to enabling an adaptive evaluation.

8.2 Future Work

There are several areas of relevance to this subject matter which will require further attention if the uptake of dynamic multi-attribute decision resolution is to be facilitated. The first of these relates to the problem of identifying and managing the trade-offs to be made between attributes, particularly if decision resolution is to become an increasingly automated task. The techniques for extracting the decision maker's preference structure, for example as described in [Kee76] need rationalizing further to enable their application to become more practicable. This topic needs to be considered from both the on- and off-line perspective. A second area of interest is to explore the hierarchy of adaptive techniques, together with alternative definitions of the notion of coverage, in order to provide some understanding as to what response is the most appropriate under a given set of circumstances. There are many ways in which adaptiveness can be enabled and exploited and, having outlined the general case, the subject may be best served by next focussing on a specific example with clearly defined boundaries before expanding to consider the subject matter further with respect to a particular domain. Finally, it would be advantageous, from a practical perspective, for engineers to have ready access to a library of evaluation methods capable of dealing with multi-attribute problems and whose properties and input requirements are known. Blithely adopting a solution involving weighting constants to tackle every problem encountered may not, necessarily, be the best approach to take.

In addition to the above issues, there is also a perceived need for further work to be carried out in the areas of human-computer interaction, particularly with respect to the psychological and subjective nature of decision resolution. For example, how do human decision makers set about identifying relevant attributes, and what are the factors which distinguish the satisfactory resolution of a decision from an unsatisfactory one? Also, how do user-preferences vary with respect to the same problem when it is encountered repeatedly over a period of time? Finally, is the inclusion of user/decision maker preferences justified in view of the additional work they generate and might the system be capable of arriving at similarly successful outcomes but using its own, more objective, devices?

8.3 Concluding Remarks

This thesis has considered issues relating to the employment of techniques for the dynamic resolution of multi-attribute decision problems. It has touched upon some, but by no means the exclusive set of, factors which may need to be considered and proposed that it is advantageous to incorporate adaptive capabilities from the outset of the definition process. Ideally, future engineers will have access to established techniques by which to model such problems and a series of tools by which to deal with them on-line. There is still some way to go, however, before this armoury of methodologies is complete.

Bibliography

[Ahi98] Ahituv N, Igabaria M & Sella A, "The Effects of Time Pressure and Completeness of Information on Decision Making", Journal of Management Information Systems, 15(2), p. 153-172, Fall 1998.

[Aky02] Akyildiz I F, Su W, Sankarasubramaniam Y & Cayirci E, "Wireless Sensor Networks: A Survey", Computer Networks, 38 (2002), p. 393-422, 2002.

[Ald99] Aldarmi S A, "Scheduling Soft-Deadline Real-Time Transactions", PhD Thesis, Department of Computer Science, University of York, UK, October 1999.

[Bar91] Baruah S, Koren G, Mao D, Mishra B, Raghunathan A, Rosier L, Shasha D & Wang F, "On the Competitiveness of On-Line Real-Time Task Scheduling", Proceedings of IEEE Real-Time Systems Symposium, San Antonio, TX, p. 106-115, December 1991.

[Bar95] Barbacci M, Klein M H, Longstaff T A & Weinstock C B, "Quality Attributes", Technical Report CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University, PA, December 1995.

[Bea] http://java.sun.com/products/javabeans

[Ber01] Bernat G & Burns A, "Three Obstacles to Flexible Scheduling", Proceedings of 13th Euromicro Conference on Real-Time Systems, Delft, Netherlands, p. 11-13, 13-15 June 2001.

[Biy88] Biyabani S R, Stankovic J A & Ramamritham K, "The Integration of Deadline & Criticalness in Hard Real-Time Scheduling", Proceedings of 9th Real-Time Systems Symposium, p. 152-160, 1988.

[Boc99a] Bocheck P, Nakajima Y & Chang S-F, "Real-time Prediction of Subjective Utility Functions for MPEG-4 Video Objects", Proceedings of PacketVideo '99, New York, NY, 26-27 April 1999.

[Boc99b] Bocheck P, Campbell A T, Chang S-F & Liao R R-F, "Utility-based Network Adaptation for MPEG-4 Systems", 9th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '99), Basking Ridge, NJ, September 1999.

[Bod94] Boddy M & Dean T, "Decision-Theoretic Deliberation Scheduling for Problem Solving in Time-Constrained Environments", Artificial Intelligence, 67(2), p. 245-285, 1994.

[Boe78] Boehm B W, Brown J R, Kaspar H, Lipow M, MacLeod G J & Merritt M J, Editors, "Characteristics of Software Quality", TRW and North Holland Publishing Company, New York, 1978.

[Bra97] Braden R, Zhang L, Berson S, Herzog & Jamin S, "Resource ReSerVation Protocol (RSVP)", The Internet Engineering Task Force RFC 2205, September 1997.

[Bur98] Burns A & Prasad D, "Value-Based Scheduling of Flexible Real-Time Systems for Intelligent Autonomous Vehicle Control", Proceedings of 3rd IFAC Symposium on Intelligent Autonomous Vehicles, Eds. Salichs M A & Halme A, Universidad Carlos III De Madrid, Spain, p. 127-132, 1998.

[Bur00] Burns A, Prasad D, Bondavalli A, Di Giandomenico F, Ramamritham K, Stankovic J & Strigini L, "The Meaning and Role of Value in Scheduling Flexible Real-Time Systems", Journal of Systems Architecture (46), p. 305-325, 2000.

[Bur01] Burns A & Wellings A, "Real-Time Systems and Programming Languages", Pearson Education Limited, Harlow, England, 3rd Edition, 2001.

[But95] Buttazzo G, Spuri M & Sensini F, "Value vs. Deadline Scheduling in Overload Conditions", Proceedings of IEEE Real-Time Systems Symposium, p. 90-95, December 1995.

[But97] Buttazzo G C, "Hard Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications", Kluwer Academic Publishers, Boston, MA, 1997.

[Cal00] Callaghan A R & Lewis K E, "A 2-Phase Aspiration-Level and Utility Theory Approach to Large Scale Design", Proceedings of ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC '00), Baltimore, MD, 10-13 September 2000.

[Cha02] Chalmers D, "Contextual Mediation to Support Ubiquitous Computing", PhD Thesis, Imperial College of Science, Technology and Medicine, University of London, UK, August 2002.

[Che96] Chen K & Muhlethaler P, "A Scheduling Algorithm for Tasks Described by Time Value Function", Real-Time Systems, 10(3), p. 293-312, 1996.

[Chu98] Chu F & Halpern J, "A Decision-Theoretic Approach to Reliable Message Delivery", Proceedings of 12th International Symposium on Distributed Computing (DISC '98), Andros, Greece, p. 89-103, September 1998.

[Cla99] Clark R, E Douglas Jensen, Kanevsky A, Maurer J, Wallace P, Wheeler T, Zhang Y, Wells D, Lawrence T & Hurley P, "An Adaptive, Distributed Airborne Tracking System", Proceedings of 7th International Workshop on Parallel and Distributed Real-Time Systems, p. 353-362, 1999.

[Coh01] Cohen N H, Purakayastha A, Turek J, Wong L & Yeh D, "Challenges in Flexible Aggregation of Pervasive Data", IBM Research Report, RC21942 (98646), IBM Research Division, Thomas J Watson Research Center, Yorktown Heights, NY, 23 January 2001.

[Cur03] Curescu C & Nadjm-Tehrani S, "Time-Aware Utility-based QoS Optimisation", Proceedings of 15th Euromicro Conference on Real-Time Systems, Porto, Portugal, p. 83-92, 2-4 July, 2003.

[Dan01] Daniels J, Werner P W & Bahill A T, "Quantitative Methods for Tradeoff Analyses", Systems Engineering, 4(3), p. 190-212, 2001.

[Dav95] Davis R I, "On Exploiting Spare Capacity in Hard Real-Time Systems", PhD Thesis, Department of Computer Science, University of York, UK, July 1995.

[Dea88] Dean T & Boddy M, "An Analysis of Time-Dependent Planning", Proceedings of 7th National Conference on Artificial Intelligence, St Paul, Minnesota, p. 49-54, 1988.

[Dea91] Dean T, "Decision-Theoretic Control of Inference for Time-Critical Applications", International Journal of Intelligent Systems, 6, p. 417-441, 1991.

[Dey01] Dey A, "Understanding and Using Context" Personal and Ubiquitous Computing, 5(1), p. 4-7, February 2001.

[Duj96] Dujmovic J J, "A Method for Evaluation and Selection of Complex Hardware and Software Systems", Proceedings of 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems (CMG 96), Vol. 1, p. 368-378, 1996.

[Dur01] Durrant-Whyte H & Stevens M, "Data Fusion in Decentralised Sensing Networks", Proceedings of 4th International Conference on Information Fusion (Fusion 2001), Montreal, CA, August 2001.

[Etz89] Etzioni O, "Tractable Decision-Analytical Control", Proceedings of First International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc, p. 114-125, 1989.

[Exp] http://www.expertchoice.com/

[Fin98] Fink E, "How to Solve it Automatically: Selection Among Problem-Solving Methods", Proceedings of 4th International Converence on Artificial Intelligence Planning Systems (AIPS '98), Carnegie Mellon University, Pittsburgh, PA, p. 128-136, 7-10 June 1998.

[Fis74] Fishburn P C, "Lexicographic Orders, Utilities and Decision Rules: A Survey", Management Science, 20(11), p. 1442-1471, July 1974.

[Fis98] Fischer S & de Meer H, "Decision Support for Cooperative QoS Management", Proceedings of 6th IEEE International Workshop on Quality of Service (IWQoS '98), Napa, CA, p. 247-255, May 1998.

[Fli99] Flinn J & Satyanarayanan M, "Energy-Aware Adaptation for Mobile Applications", Proceedings of 17th ACM Symposium on Operating Systems Principles, Kiawah Island, SC, p. 48-63, December 1999.

[Gar03] Garlan D, Cheng S & Schmerl B, "Increasing System Dependability through Architecture-Based Self-Repair" in Architecting Dependable Systems, de Lemos R, Gacek C and Romanovsky A Editors, LNCS 2677, Springer-Verlag Publishers, 2003.

[Gho04] Ghosh S, "Scalable QoS-based Resource Allocation", PhD Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, PA, August 2004.

[Gil05] Gilb T & Maier M W, "Managing Priorities: A Key to Systematic Decision Making", INCOSE International Symposium, Rochester, NY, USA, 10-15 July 2005.

[Hal04] Hallsteinsen S, Stav E & Floch J, "Self-Adaptation for Everyday Systems", Workshop on Self-Managing Systems (WOSS '04), Newport Beach, CA, 31 Oct - 1 November 2004.

[Hol98] Holtman K & Mutz A, "HTTP Remote Variant Selection Algorithm -- RVSA/1.0", Internet Engineering Task Force RFC 2296, March 1998.

[Hor87] Horvitz E J, "Reasoning about Beliefs and Actions under Computational Resource Constraints", Proceedings of 3rd Workshop on Uncertainty in Artificial Intelligence, Seattle, WA, 1987.

[Hor88] Horvitz E J, "Reasoning Under Varying and Uncertain Resource Constraints", Proceedings of 7th National Conference on Artificial Intelligence, Minneapolis, MN, p. 111-116, 1988.

[IBM05] IBM, "An Architectural Blueprint for Autonomic Computing", White Paper 3rd Edition, downloaded from: http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf, June 2005.

[ISO98] International Organization for Standardization, "Information Technology - Quality of Service: Framework", ISO/IED 13236:1998, 1998.

[ITU98] International Telecommunications Union (ITU), "Methodology for the Subjective Assessment of the Quality of Television Pictures", Recommendation BT.500.10, 2000.

[Jen85] Jensen E D, Locke C D & Tokuda H, "A Time-Driven Scheduling Model for Real-Time Operating Systems", Proceedings of 6th IEEE Real-Time Systems Symposium, p. 112-122, San Diego, CA, 3-6 December 1985.

[Kat94] Katz R H, "Adaptation and Mobility in Wireless Information Systems", IEEE Personal Communications, 1(1), p. 6-17, First Quarter 1994.

[Kee76] Keeney R L & Raiffa H, "Decisions with Multiple Objectives: Preferences and Value Tradeoffs", John Wiley & Sons, Inc., New York, 1976.

[Kel03] Kelly T, "Utility-Directed Allocation", Proceedings of 1st Workshop on Algorithms and Architectures for Self-Managing Systems, San Diego, CA, June 2003.

[Kep03] Kephart J O & Chess D M, "The Vision of Autonomic Computing", IEEE Computer Society, Computer, p. 41-50, January 2003.

[Kha97] Khan S, Li K F & Manning E, "The Utility Model for Adaptive Multimedia Systems", Proceedings of 4th International Conference on Multimedia Modelling, Singapore, 17-20 November 1997.

[Kim01] Kim J et al, "Collective Value of QoS: A Performance Measure Framework for Distributed Heterogeneous Networks", 10th Heterogeneous Computing Workshop (HCW 2001) in Proceedings of 15th International Parallel & Distributed Processing Symposium, San Francisco, CA, 23-27 April 2001.

[Kon95a] Kontio J, "OTSO: A Systematic Process for Reusable Software Component Selection", Department of Computer Science, University of Maryland, Technical Report CS-TR-3478, December 1995.

[Kon95b] Kontio J, Chen S, Limperos K, Tesoriero R, Caldiera G & Deutsch M, "A COTS Selection Method and Experiences of its Use", Proceedings of 20th Annual Software Engineering Workshop, NASA Software Engineering Laboratory, Greenbelt, MD, 1995.

[Kra99] Krasic C & Walpole J, "QoS Scalability for Streamed Media Delivery", CSE-99-011, Department of Computer Science, Oregon Graduate Institute of Science & Technology, Portland, OR, 1999.

[Kuj03] Kujawski E, "Multi-Criteria Decision Analysis: Limitations, Pitfalls, and Practical Difficulties", Proceedings of 13th Annual International Symposium INCOSE 2003, Washington, DC, p. 1169-1176, 1-3 July 2003.

[Lag00] Lagoudakis M G & Littman M L, "Algorithm Selection using Reinforcement Learning", Proceedings of 17th International Conference on Machine Learning, Stanford University, CA, p. 511-518, 29 June-2 July 2000.

[Lar04] Larsson M, "Predicting Quality Attributes in Component-Based Software Systems", PhD Thesis, Department of Computer Science and Engineering, Malardalen University, Vasteras, Sweden, March 2004.

[Lee99] Lee C, "On Quality of Service Management", PhD Thesis, School of Computer Science, Carnegie Mellon University, PA, August 1999.

[Leg04] Legg G, "ZigBee: Wireless Technology for Low-Power Sensor Networks", downloaded from: http://www.techonline.com/community/ed_resource/feature_article/36561, 6 May 2004.

[Li99] Li B & Nahrstedt K, "Dynamic Reconfiguration for Complex Multimedia Applications", Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS '99), p. 165-170, June 1999.

[Lia01] Liao R R-F & Campbell A T, "A Utility-Based Approach for Quantitative Adaptation in Wireless Packet Networks", Wireless Networks, 7, p. 541-557, 2001.

[Liu73] Liu C & Layland J, "Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment", Journal of the ACM, 20(1), p. 46-61, 1973.

[Loc86] Locke C D, "Best-Effort Decision Making for Real-Time Scheduling", PhD Thesis, CMU-CS-86-134, Department of Computer Science, Carnegie Mellon University, 1986.

[Mai02] Mainwaring A, Polastre J, Szewczyk R, Culler D & Anderson J, "Wireless Sensor Networks for Habitat Monitoring", Proceedings of 1st International Workshop on Wireless Sensor Networks and Applications (WSNA '02), Atlanta, Georgia, p. 88-97, 28 September 2002.

[Mar00] Marmasse N & Schmandt C, "Location-Aware Information Delivery with ComMotion", Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC 2000), Bristol, UK, p. 157-171, September 2000.

[MET] http://www.metoffice.gov.uk/climate/uk/

[Ncu02] Ncube C & Dean J C, "The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components", Proceedings of 1st International Conference on COTS-Based Software Systems (ICCBSS 2002), Orlando, FL, p. 176-187, February 2002.

[Net] http://www.microsoft.com/net/default.mspx

[Neu47] von Neumann J & Morgenstern O, "Theory of Games and Economic Behaviour", 2nd Edition, Princeton University Press, Princeton, MA, 1947.

[Nob97] Noble B D, Satyanarayanan M, Narayanan D, Tilton J E, Flinn J & Walker K R, "Agile Application-Aware Adaptation for Mobility", Proceedings of 16th ACM Symposium on Operating Systems Principles, Operating System Review, 31(5), St Malo, France, p. 276-287, 5-8 October 1997.

[Nob00] Noble B, "System Support for Mobile, Adaptive Applications", IEEE Personal Communications, p. 44-49, February 2000.

[Oli02] Oliver I, "Quality of Service Specification in Dynamically Replaceable Component Based Systems", Component Based Software Engineering Workshop, 9th IEEE Conference and Workshops on Engineering of Computer-Based Systems, Lund, Sweden, 8-11 April, 2002.

[Ore99] Oreizy P, Gorlick M M, Taylor R N, Heimbigner D, Johnson G, Medvidovic N, Quilici A, Rosenblum D S & Wolf A L, "An Architecture-Based Approach to Self-Adaptive Software" IEEE Intelligent Systems, 14(3), p. 54-62, May/June 1999.

[Ozt03] Ozturk M, Tsoukias A & Vincke P, "Preference Modelling", DIMACS Technical Report 2003-34, DIMACS, Rutgers University, NJ, October 2003.

[Par01] Park J-T & Baek J-W, "Management of Service Level Agreements for Multimedia Internet Service Using a Utility Model", IEEE Communications Magazine, p. 100-106, May 2001.

[Pay96] Payne J W & Bettman J R, "When Time is Money: Decision Behaviour under Opportunity-cost Time Pressure", Organizational Behaviour and Human Decision Processes, 66(2), p. 131-152, May 1996.

[Pol02] Poladian V, Garlan D & Shaw M, "Software Selection and Configuration in Mobile Environments: A Utility-based Approach", Position Paper, 4th International Workshop on Economics-driven Software Engineering, Orlando, FL, 21 May, 2002.

[Pol04] Poladian V, Sousa J P, Garlan D & Shaw M, "Dynamic Configuration of Resource-Aware Services", Proceedings of 26th International Conference on Software Engineering (ICSE), Edinburgh, Scotland, p. 604-613, May 2004.

[Poy97] Poyhonen M & Hamalainen R P, "On the Convergence of Multiattribute Weighting Methods", Decision Analysis Society Working Paper WP980008, downloaded from: http://fisher.osu.edu/~butler_267/DAPapers/WP980008.html, April 1997.

[Pra98] Prasad D K, "Dependable Systems Integration using Measurement Theory and Decision Analysis", PhD Thesis, Department of Computer Science, University of York, UK, November 1998.

[Pra03] Prasad D, Burns A & Atkins M, "The Valid Use of Utility in Adaptive Real-Time Systems", Real-Time Systems, 25, p. 277-296, 2003.

[Raf02] Raffo D M, Harrison W & Vandevill J, "Software Process Decision Support: Making Process Tradeoffs Using a Hybrid Metrics, Modeling and Utility Framework", Proceedings of IEEE 14th International Conference on Software Engineering and Knowledge Engineering (SEKE '02), Ischia, Italy, p. 803-809, 15-19 July 2002.

[Raj98] Rajkumar R, Lee C, Lehoczky J P & Siewiorek D P, "Practical Solutions for QoS-based Resource Allocation Problems", Proceedings of IEEE Symposium on Real-Time Systems, p. 296-306, Madrid, December 1998.

[Ram84] Ramamritham K & Stankovic J A, "Dynamic Task Scheduling in Distributed Real-Time Systems", IEEE Software, 1(3), p. 65-75, July 1984.

[Rav98] Ravindran B, Welch L R, Bruggeman C, Shirazi B A & Cavanaugh C, "A Resource Management Model for Dynamic Scalable, Dependable, Real-Time Systems", Proceedings of 12th International Paralledl Processing Symposium (IPPS)/Symposium on Parallel and Distributed Processing (SPDP) Workshop Program, Orlando, FL, p. 931-936, 30 March - 3 April 1998.

[Ric99] Richardson P & Sarkar S, "Adaptive Scheduling: Overload Scheduling for Mission Critical Systems", Proceedings of Real-Time Technology and Applications Symposium, Vancouver, BC, p. 14-23, 2-4 June 1999.

[Ric01] Richardson P, Sieh L & Ganz A, "Quality of Service Support for Multimedia Applications in Third Generation Mobile Networks Using Adaptive Scheduling", Real-Time Systems, 21(3), p. 269-284, 2001.

[Roy87] Roy B & Bouyssou D, "Famille de critères: problème de cohérence et de dépendance", Document du Lamsade 37, Université Paris-Dauphine, 1987.

[Rus89] Russell S & Wefald E, "Principles of Metareasoning", Proceedings of 1st International Conference on Principles of Knowledge Representation and Reasoning, R J Brachman et al Editors, Morgan Kaufman, Los Altos, CA, p. 400-411, 1989.

[RWM05] The US Department of Transport Federal Highway Administration Road Weather Management Program, details available at: http://www.ops.fhwa.dot.gov/weather/mitigating_impacts/surveillance.htm, accessed 28 September 2005.

[Saa92] Saaty T, "Multicriteria Decision Making - The Analytical Hierarchy Process", Technical Report, University of Pittsburgh, RWS Publications, 1992.

[She95] Shenker S, "Fundamental Design Issues for the Future Internet", IEEE Journal of Selected Area in Communication, 13(7), p. 1176-1188, 1995.

[Sim86] Simon H A et al, "Report of the Research Briefing Panel on Decision Making and Problem Solving", National Academy of Sciences, National Academy Press, Washington, DC, p. 19-35, 1986.

[Sta93] Stankovic J A & Ramamritham K Editors, "Advances in Real-Time Systems", IEEE Computer Society Press, Los Alamitos, CA, 1993.

[Ste00] Steere D C, Baptista A, McNamee D, Pu C & Walpole J, "Research Challenges in Environmental Observation and Forecasting Systems", Proceedings of 6th International Conference on Mobile Computing and Networking (Mobicom '00), Boston, MA, p. 292-299, 6-11 August 2000.

[Thi04] Thiele L & Wilhelm R, "Design for Timing Predictability", Real-Time Systems, 28(2-3), p. 157-177, November 2004.

[Til02] Tilak S, Abu-Ghazaleh N B & Heinzelman W, "Infrastructure Tradeoffs for Sensor Networks", 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02) in Conjunction with ACM MobiCom 2002, Atlanta, GA, September 2002.

[Tri95] Triantaphyllou E & Mann S H, "Using the Analytic Hierarchy Process for Decision Making in Engineering Applications: Some Challenges", International Journal of Industrial Engineering, 2(1), p.35-44, 1995.

[Van99] Vandalore B, Feng W, Jain R & Fahmy S, "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia", OSU Technical Report OSU-CISRC-5/99-TR14, Department of Computer and Information Science, The Ohio State University, Columbus, OH, 1999.

[Vin92] Vincke P, "MultiCriteria Decision-Aid", John Wiley & Sons Ltd, Chichester, England, 1992.

[Wal04] Walsh W E, Tesauro G, Kephart J O & Das R, "Utility Functions in Autonomic Systems", Proceedings of IEEE 1st International Conference on Autonomic Computing (ICAC '04), New York, NY, p.70-77, May 2004.

[Wan99] Wang X & Schulzrinne H, "Comparison of Adaptive Internet Multimedia Applications", IEICE Transactions on Communications, E82-B, p. 806-818, June 1999.

[Wan00] Wang X & Schulzrinne H, "Adaptive Reservation: A New Framework for Multimedia Adaptation", Proceedings of IEEE International Conference on Multimedia and Expo (ICME), New York, NY, p. 1051-1054, 30 July - 2 August 2000.

[Wan04] Wang J & Ravindran B, "Time-Utility Function-Driven Switched Ethernet: Packet Scheduling Algorithm, Implementation, and Feasibility Analysis", IEEE Transactions on Parallel and Distributed Systems, 15(1), p. 119-133, January 2004.

[Wei91] Weiser M, "The Computer for the Twenty-First Century" Scientific American, 265(3), p. 94-104, September 1991.

[Wel01] Welch L R & Brandt S, "Toward a Realization of the Value of Benefit in Real-Time Systems", Proceedings of 15th International Parallel & Distributed Processing Symposium, San Fransciso, CA, p. 30093A, 23-27 April 2001.

[Whi76] White D J, "Fundamentals of Decision Theory", North-Holland Publishers, New York, 1976.

[Wu01] Wu D, Hou Y T, Zhu W, Zhang Y & Peha J M, "Streaming Video over the Internet: Approaches and Directions", IEEE Transactions on Circuits and Systems for Video Technology, 11(1), p. 282-300, February 2001.

[Yau04] Yau S S & Karim F, "An Adaptive Middleware for Context-Sensitive Communications for Real-time Applications in Ubiquitous Computing Environments", Real-Time Systems, 26(1), p. 29-61, January 2004.

[Yel03] Yellin D M, "Competitive Algorithms for the Dynamic Selection of Component Implementations", IBM Systems Journal, 42(1), p. 85-97, 2003.