# The Valid Use of Utility in Adaptive Real-Time Systems*

D. Prasad (divya@mca-ltd.com)
*Mission Critical Applications Limited, York, U.K.*

A. Burns (burns@cs.york.ac.uk)
*Real-Time Systems Research Group, Department of Computer Science, University of York, U.K.*

M. Atkins (martin@mca-ltd.com)
*Mission Critical Applications Limited, York, U.K.*

**Abstract.** Real-time systems that must adapt their behaviour to changes in internal and external circumstances require flexibility in their scheduling. One approach that has been advocated for achieving this flexibility is called 'value-based' scheduling, wherein services are distinguished based upon their current utility values. The main result of this paper is that the assumptions used in the assignment of these values must be matched with the way the values are used in scheduling. The key notion for ensuring this is the 'scale type' of the value, in the sense defined by measurement theory. There are simple tests for 'meaningful' uses of values based on their scale types, and we apply these tests to investigate the meaningfulness requirements of some commonly-used scheduling approaches.

**Keywords:** measurement, scales, value, utility, adaptive systems, flexible scheduling

## 1. Introduction

Most present real-time systems are scheduled statically, i.e., a fixed set of *services* is mapped onto the available resources, and pre-run-time checks are used to ensure that all timing constraints are satisfied. We use the term *service* to refer to a task, or a group of tasks that implement a function. In contrast, the *adaptive* real-time systems of the future will need more flexibility in the run-time phase, to accommodate dynamic information in their operations. They must modify the set of services they support, and the extent of their functions, in accordance with environmental and internal demands. This type of run-time flexibility can be used to enhance the quality of service of the system by providing the best possible level of functionality that can be achieved in the prevailing circumstances (given fixed amounts of computational and communication resources in the system).

Recent scheduling literature has advocated approaches that provide this type of run-time flexibility by distinguishing between services in terms of their 'utility'. These approaches are generally referred to as *value-based* (or *reward-based*) scheduling techniques. There are two issues that need to be resolved for value-based scheduling to be practicable: how to *assign* values, and how to *use* values. Most existing literature (for example, Locke (1986), Wendorf (1988), Tokuda et al. (1987), Davis et al. (1995)) has concentrated on the second problem in isolation, and little has been said about how to assign values, until recent work by the authors (Burns et al., 2000; Prasad and Burns, 2000; Burns and Prasad, 1998; Prasad, 1998). We have previously conjectured that the two problems have inter-dependent solutions (Prasad, 1998).

In this paper, we investigate the theoretical link between the assignment and use of values. Earlier publications (Burns et al., 2000; Prasad and Burns, 2000) have addressed some of the issues involved

* This work was started in the Real-Time Systems Research Group at the University of York, U.K. under the sponsorship of the UK EPSRC (Project Number GR/L73456).

in value assignment; here we concentrate on the link with the use of value in scheduling. Sections 2 and 3 provide the background and context for value-based scheduling in adaptive real-time systems. Section 4, which is the spine of the paper, first outlines the necessary formalism from the theory of measurement, and then applies it to the concept of value. The notions of scale types and meaningfulness are described and shown to be key to establishing the theoretical link between the measurement and usage of value. Section 5 concludes the paper.

## 2. The Need for Run-Time Decisions in Adaptive Systems

For systems that must operate in, and adapt to, a changing environment, some decisions are best postponed until after the delivery or deployment of the system — indeed into run-time. Leaving some decisions as late as possible allows the behaviour of the system to be better tailored to the dynamic conditions of the environment. The motivation for dynamic schedules (incorporating run-time decisions) is based on two observations about the behaviour of typical static schedules:

1. **Inefficient Resource Usage:** The static schedules have to be pessimistic in predicting resource requirements (computation and communication resources, which are the main costs in real-time systems). Static schedules ensure that there are sufficient resources to cope with the maximum possible load on the system (i.e. worst-case execution times, worst possible phasings, and worst-case arrival of sporadic work), and consequently, the average resource utilisation in the run-time system is low. There is considerable wasted spare capacity that could have been used for running 'value added' services.

2. **Non-Graceful Degradation:** Static schedules react inflexibly to failures and overloads. A static scheme may be able to cope with certain failures (as defined in its failure model), but once the system moves outside its failure model or load hypothesis, there is no guaranteed minimum level of service that can be depended upon. Hence there is a need to facilitate graceful degradation of services when resources become scarce.

Both of these limitations could potentially be overcome by allowing run-time decisions to influence the scheduling of the services. These decisions would involve identifying the extra services to be supported when spare resources are available, or identifying the services to be sacrificed when resources are scarce. However, postponing these decisions until run-time would be worthwhile only if:

- It would result in a less pessimistic resource usage.
- Knowledge of the run-time data and conditions enables better decisions to be made about the set of services that should be supported or sacrificed.
- The overheads involved in making run-time decisions do not outweigh the potential benefits. This is a critical issue which has often been overlooked in published research (e.g. (Jensen et al., 1985)).

A further complication is that due to limited resources, it may not be possible to support all services at all times, or to provide maximum functionality in the services that *are* supported. In addition, all services do not provide equal utility to the system at all times, and they do not have equal cost. This is why we need some measure of utility to decide which subset of services should be supported at any instant of time. This topic is pursued further in Section 3.

### 2.1. Representation of Services and Alternatives

We consider a model of the run-time behaviour of a real-time system which consists of a set of *services*. Each service may be realised by one or more *service alternatives* — typically only one of which is needed for any particular invocation of the service. In the case of some services, it may be possible to enhance overall system utility by running more than one alternative. Services are either periodic or sporadic and there is an upper bound on the number of possible invocations within any time interval. The entire system is considered to run indefinitely, hence over the entire lifetime of the system each service has an unbounded number of invocations. Let $S$ represent the system consisting of a set of $n$ services:

$$S = \{S_1, S_2, \ldots, S_n\}$$

Each service is composed of one or more (say $m_i$) alternatives:

$$S_i = \{A_i^1, A_i^2, \ldots, A_i^{m_i}\}$$

In some applications we may only wish to compare the alternatives $A_i^j$ within a particular service $S_i$. In other applications we may only need to compare the services $S_i$ with each other rather than comparing the alternatives within each service. Hereafter, we will simply use $A$ to denote the current set of alternatives (whatever combination of $S_i$ and $A_k^j$ they might represent) and the notation $A_i$, $A_j$, and so on to denote the elements of $A$.

### 2.2. An Example: Autonomous Vehicle Systems

A good example of the sorts of systems that will need the adaptability described in this paper is a real-time control system for autonomous vehicles of the future. Such systems will have to exhibit intelligent behaviour in order to function in the highly dynamic and non-deterministic environment within which they are deployed. This environment is characterised by the unpredictable nature of other vehicles, obstructions, route information, weather and road or terrain conditions. Furthermore, the consequences of system failure due to faulty software, hardware, or sensors may be catastrophic. Hence, autonomous vehicle control systems must be resilient to such failures and have the property of graceful degradation under conditions of overload. Finally, cost and space constraints dictate that the most effective use must be made of the limited processing and communication resources available on board.

Table I presents an example list of services and service alternatives, similar to that in (Burns and Prasad, 1998). It shows the functionality of the system divided into seven sub-systems (i.e. services), each with a number of service alternatives that implement different algorithms. We assume a software/hardware architecture similar to that proposed for Integrated Modular Avionics (AEEC, 1991) where all software components share a common set of computing resources, rather than a federated system of 'black-box' components. Hence the services and alternatives in Table I are assumed to be all competing for the same resources.

Table I. Autonomous Vehicle Example: Services and Service Alternatives

| SERVICES | SERVICE ALTERNATIVES |
|---|---|
| Collision Prevention | Infra-Red Beam Deflection (IRBD)<br>RADAR<br>Short Range Communication (SRC)<br>Stereo Vision based vehicle Tracking (SVT) |
| Braking Control | Basic Braking System (BBS)<br>Anti-lock Braking System (ABS)<br>Load Sensitive Anti-Lock Braking (LSALB) |
| Engine Control | Basic Control (BC)<br>Increased Precise Computation(1) (IPC1)<br>Increased Precise Computation(2) (IPC2) |
| Lateral Control (input processing) | Magnetic Markers (MM)<br>Line Following (LF) |
| Lateral Control (output) | Proportional Integral Derivative (PID)<br>Frequency Shape Linear Quadratic (FSLQ)<br>Fuzzy Rule-based (FR)<br>Neuro Control (NC) |
| Path Finding | Basic Data Fusion (BDF)<br>Fuel Optimisation (FO)<br>Ride Comfort (RC)<br>Time Optimisation (TO) |
| Route Planning | Middle Level Planning (MLP)<br>GPS-based global planning (GPS) |
| Displays Control | Basic Update Rates (BUR)<br>Speed Warnings (SW)<br>Fuel Consumption Analysis (FCA) |

## 3. The Role of Utilities and Values

The traditional static approach to scheduling (Burns and Wellings, 2001) uses only well-known attributes of the services such as deadline $D$, period $T$ (for periodic services) or minimum inter-arrival time (for non-periodic services), and worst case execution time $C$. More recent real-time systems literature (Jensen et al., 1985; Locke, 1986; Buttazzo and Stankovic, 1993; Davis et al., 1995) has drawn attention to an attribute called the *value* ($V$) of a service[1], which is a measure of the intrinsic benefit obtained by running that particular service. Many of the run-time details useful for adaptive behaviour can be encompassed under the umbrella of value, and this can be used to drive the run-time decision-making process.

[1] The use of the term 'value' is unfortunate since this work considers issues of timing, rather than data values. However this is standard terminology in this area, and the reader should not confuse the worth (value) of a service or task, with the result (*value*) it delivers.

### 3.1. Value-based Scheduling

The decision problem in value-based scheduling is to choose which services to execute to maximise the utility of the system, while meeting a number of conflicting resource constraints and application requirements. A poorly made decision might cause a vital service to be delayed (or even to not be executed), possibly leading to catastrophic consequences. We will see that a number of properties must be taken into account in making such decisions. We keep track of at least four distinct, and potentially conflicting properties, all of which affect the Quality of Service (QoS) provided by the system:

- Real-time behaviour: Decisions must be consistent with (must not cause a violation of) the timing guarantees that are necessary to ensure the real-time performance of the system.
- Resource constraints: The choice between services is dictated primarily by the limited computational resources available (and the varying costs of different services). If resources were not limited, the decision problem would not occur, because all the desired services could always be executed.
- Dependability: Many applications demand that properties such as safety, reliability and availability are not compromised in making run-tim decisions.
- Functionality: The important functional requirements of the application must be fulfiled as far as possible, even in degraded modes of operation.

We cannot make decisions that favour any of the above properties in isolation, because they are all inter-related, and must often be traded-off against each other. Being a run-time decision, it also becomes important that a 'good' (acceptable) decision can be made quickly and within the available resources. Thus we will also have to trade-off the effort spent in making the decision against the quality of the decision, and the time available to carry out the results of the decision.

*Value-based scheduling* is thus a decision problem involving a choice of a subset of services from the set of services currently available for execution. This decision has to be revisited frequently so that the 'best possible' outcome ensues. Some of the available services are hard and must be executed under normal operational circumstances, and hence do not figure in the decision. Other services add value to the system when they are run, but are not essential to the operation of the system, and hence can be considered for value-based scheduling.

Unfortunately, some of the factors that contribute to value become known only at run-time, and moreover, may change over time. Hence the value of a service cannot easily be calculated in advance. On the other hand, there are considerable computational overheads (Burns et al., 2000) involved in computing values on-line.

Typically, the value of a service is dependent upon the quality of its output, the time at which it completes, the recent history of the system, the current state of the computational resources and the system's environment. The cost of providing the service is, of course, also important. Furthermore, a probabilistic element could be introduced by considering the likelihood of completing an alternative by its deadline.

### 3.2. Policies for Using Value in Run-time Decision Making

This section surveys some commonly advocated approaches for using the concept of value to provide flexible run-time behaviour. There are many issues concerning the use of value at run-time. An important one is to decide if value will be used for actually scheduling the service alternatives (e.g. to modify the dynamic priorities of the alternatives within a priority-based scheduling scheme (Burns and Wellings, 2001)), or as a mechanism for deciding which service alternatives are to be admitted for scheduling until the next decision-point. As value-based scheduling is a resource-constrained decision problem (Burns et al., 2000), it is necessary to know the following information for every service (and alternative within a service), in order to implement any scheduling or admission policy: value $V$, resource requirement $C$(cost, typically worst-case execution time), inter-arrival time $T$ and deadline $D$.

The most straightforward scheduling approach is to use $V$ directly and in isolation (Biyabani et al., 1988; Jensen et al., 1985), i.e. highest value alternative first — thus equating value with run-time priority. An alternative approach is to use value as an admission policy, with the actual scheduling of the admitted alternatives being undertaken by a classical scheduling technique such as Earliest Deadline First (EDF), or fixed priority preemptive scheduling, where priorities are derived from $T$ or $D$.

Other schemes have not used value directly but have incorporated it into another parameter; e.g. value-density, $\Omega$ (Locke, 1986) or valueness-deadline, $\Lambda$ (Huang et al., 1989):

$$\Omega = V/C \tag{1}$$

$$\Lambda = V/D \tag{2}$$

with $\Omega$ or $\Lambda$ being used either to define scheduling priority or as part of an admissions policy.

The evidence from several studies (Locke, 1986; Buttazzo and Stankovic, 1993; Davis et al., 1995) indicates that there are some advantages in having an admission policy that uses value, either directly or within another parameter such as $\Omega$, and having EDF as the scheduling policy.

Another value-based admissions policy uses the notion of *threshold* (Davis, 1995). Here an alternative is only admitted if its value (or $\Omega$ or $\Lambda$) is higher than the average value (or $\Omega$ or $\Lambda$) achieved by recently completed alternatives.

If $\Omega$ or $\Lambda$ (or some derivative of these schemes) is used then the definition of value must itself be independent of $C$ or $D$. Alternatives have different utility and different cost (in terms of run-time resources). Hence, either 'value' should incorporate 'cost', in which case the run-time decision process must not incorporate it, or else, 'value' and 'cost' should be passed to the run-time scheduler independently. Thus we see that the two parts of the problem — assigning values and using them — are not independent.

### 3.3. Use of Values in Autonomous Vehicle Example

Let us re-visit the autonomous vehicle example to make these ideas more concrete. In Table II we consider only a subset of the full set of services and alternatives that were shown in Table I (in fact, this subset represents the non-mandatory, value added services that could be considered for value-based scheduling under normal operating conditions). The periods ($T$) and estimates of the worst-case execution times ($C$) have been added.

The table also shows numbers for value ($V$) which were generated using a value assignment method as described in earlier publications by the authors (Prasad, 1998; Burns et al., 2000). They represent the results of the value calculation for one set of environmental and system conditions (day-time visibility, dry road conditions, and a fault-free system). The last column shows the value density ($\Omega$), calculated using Equation 1 (Section 3.2).

Table II. Autonomous Vehicle Example Showing $T$, $C$, $V$ and $\Omega$

| Service | Name | Period ($T$) | WCET ($C$) | Value ($V$) | ($\Omega$) |
|---|---|---|---|---|---|
| Collision Prevention | SVT | 100 ms | 60 ms | 59 | 0.98 |
| | SRC | 100 ms | 10 ms | 39 | 3.9 |
| Braking Control | ABS | $\leq$ 50 ms | 20 ms | 183 | 9.15 |
| | LSALB | $\leq$ 50 ms | 5 ms | 219 | 43.8 |
| Engine Control | IPC1 | 50 $\mu$s | 10 $\mu$s | 96 | 9.6 |
| | IPC2 | 50 $\mu$s | 20 $\mu$s | 19 | 0.95 |
| Lateral Control (input processing) | LF | 200 ms | 70 ms | 68 | 0.97 |
| Lateral Control (output) | FSLQ | 50 ms | 20 ms | 46 | 2.3 |
| | FR | 50 ms | 15 ms | 27 | 1.8 |
| | NC | 50 ms | 30 ms | 23 | 0.77 |
| Path Finding | FO | 1 s | 200 ms | 55 | 0.28 |
| | RC | 1 s | 300 ms | 41 | 0.14 |
| | TO | 1 s | 300 ms | 27 | 0.09 |
| Route Planning | MLP | 10 – 100 s | 3 s | 26 | 0.01 |
| | GPS | 10 – 100 s | 50 ms | 16 | 0.32 |
| Displays Control | SW | 50 ms | 50 $\mu$s | 42 | 0.84 |
| | FCA | 50 ms | 20 $\mu$s | 14 | 0.7 |

The range of numbers for $V$ (from 14 to 219) suggests that there may be significant differences between the utilities of the services, and that value-based scheduling is worthwhile. Also, only a small number of services can really be scheduled, given the values of $T$ and $C$ (since the total utilisation $U$ of resources, which is defined as $U = \Sigma C_i/T_i$ should not be more than 100% — in this case, if all the services had to be run, it would be approximately 460%), so it is crucial that the right subset of services are chosen.

We have seen that there are several different ways of using value described in the literature: directly, via some formula, or using some notion of average. All value-based scheduling schemes rely on the correctness of the numbers generated by the value assignment method. However, the literature does not recognise that these schemes also need to be consistent with certain assumptions that might have been made implicitly during the assignment of values. The next section investigates these assumptions and discusses their role in determining the appropriate use of the values in various scheduling schemes.

## 4. Scales of Measurement and their Significance

This section briefly describes the *axiomatic* or *representational* theory of measurement (Roberts, 1979; Krantz et al., 1971), based on sets, relations, axioms, and functions. We formalise the notions of utility and value in terms of measurement theory and its scales, and then discuss how values can be assigned and used in a manner that is consistent with the theory.

## 4.1. Measurement Theory Fundamentals

*Measurement* is defined as the process by which numbers or symbols are assigned to attributes of entities in the real world in order to describe them clearly. An *entity* is an object or an event (a thing) and an *attribute* is a feature or property of the entity we are interested in. In the context of this paper, entities are services (and their alternatives), whereas attributes are criteria that contribute to utility.

A *measure* is an empirical objective assignment (mapping) of a number or symbol to an entity to characterise a specific attribute. It is also noteworthy that the word *metric* is often confusingly used to refer to a measure or even an attribute. A more commonly accepted usage of the term metric (Fenton, 1991) is that of a proposed measure. For any given attribute, empirical observations about the relationships between the chosen set of entities are expressed as binary (and in general as n-ary) relations. The entities themselves must conform to some appropriate model (e.g. the computational model of what constitutes a service or a service alternative) so that the empirical relations are well-defined.

Fenton (Fenton, 1991) observes that there are four questions to which the answers must be known for any measurement activity to be theoretically sound:

1. When are we ready to make measurements? In other words, how much do we need to know about some attribute before it is reasonable to consider measuring it? For example, do we know enough about utility to be able to measure it?

2. How do we know if we have truly captured an attribute by a certain measure? For example, does the count of 'missed deadlines' measure the unreliability of a real-time system, and if not, what does it really measure?

3. Using measurements, what kinds of statements can we make about an attribute and the entities that possess it, that are meaningful? For example, is it meaningful to talk about a 20% increase in value?

4. What are the meaningful operations that we can perform on measures? For example, can we compute the average utility of a set of services?

One does not explicitly worry about these questions in carrying out classical measurements such as those of mass, length or time because our knowledge of these attributes, accumulated over many centuries, includes the answers to these questions. This is why we now have sophisticated tools for measuring these attributes. On the other hand, attributes of computer systems, are not as well understood. It is especially important to address these four questions for a concept such as utility that is relatively new within the context of real-time systems and their scheduling. We will see that Measurement Theory offers answers to the first two questions through the concept of the 'representation condition', and to the last two questions through the notion of 'scales' of measurement.

### 4.1.1. *Empirical and Numerical Relation Systems*

For a given entity and an attribute of interest, our empirical observations and/or our intuitive understanding of the attribute can be captured in terms of n-ary relations and axioms on them (Roberts, 1979; Krantz et al., 1971; Fenton, 1991). In general, the attribute is characterised by a set of empirical relations $\mathcal{R}$. This set together with the set of entities $A$ is called the *empirical relation system*, and is denoted by $\mathbf{E} = (A, \mathcal{R})$. The set $\mathcal{R}$ may be enriched over a period of time by adding new relations reflecting our improved understanding of the attribute.

Having constructed an empirical relation system for an attribute, we next need to find a *number system* into which to map our entities. A *numerical relation system*, $\mathbf{N} = (N, \mathcal{P})$ consists of the set of numbers, $N$, together with a set $\mathcal{P}$ of numerical relations, such that for each empirical relation, there is a 'corresponding' numerical relation. The formal meaning of the correspondence is explained through the *representation condition* described below. $\mathbf{M}$ is said to be a measure for an attribute if it is a mapping from $\mathbf{E} = (A, \mathcal{R})$ (for the attribute) to $\mathbf{N} = (N, \mathcal{P})$. If in addition, $\mathbf{M}$ *preserves* the relations, it is a representation or a homomorphism. The representation condition requires that under the mapping $\mathbf{M}$, correspondence is established between the entities in $A$ and the numbers in $N$ in such a way that relations in $\mathcal{R}$ *imply and are implied by* relations in $\mathcal{P}$. The representation condition is a formal definition of what is meant by a true measure that captures the attribute as intended.

As a simple example of $\mathbf{E}$ and $\mathbf{N}$ consider a a set of objects $A = \{Obj_1, Obj_2, Obj_3, Obj_4\}$, and the empirical observation 'longer than'. Imagine that we are living in a primitive age, with no well understood scales for measuring length (such as the imperial or the metric). In this situation, we might attempt to capture our understanding of the attribute 'length' by placing the objects side by side, and making observations of the following kind:

$$\begin{aligned}\mathcal{R} = \{\ & (Obj_1 \text{ is longer than } Obj_2),\\ & (Obj_3 \text{ is longer than } Obj_2),\\ & (Obj_4 \text{ is longer than } Obj_1),\\ & (Obj_3 \text{ is longer than } Obj_4)\}\end{aligned}$$

where "is longer than" is an empirical observation made by a particular observor.

A useful number system would be the set of integers $\mathbf{N}$, with $\mathcal{P}$ being the numerical relation $>$.

Then we could define a homomorphic mapping as follows:

$$\mathbf{M} = (\{(Obj_1 \mapsto 2), (Obj_2 \mapsto 1), (Obj_3 \mapsto 4), (Obj_4 \mapsto 3)\}, \mathcal{R} \mapsto >)$$

This mapping is homomorphic because it preserves all the relationships implied by $(A, \mathcal{R})$ and by $(N, \mathcal{P})$.

Another equally valid homomorphism might be:

$$\mathbf{M} = (\{(Obj_1 \mapsto 119), (Obj_2 \mapsto 50), (Obj_3 \mapsto 212), (Obj_4 \mapsto 209)\}, \mathcal{R} \mapsto >)$$

An infinite number of other possible homomorphisms exist, and any of them would be equally valid measurement mappings. Of course, for an arbitrarily large number of assorted objects (some of which may be difficult to place beside other objects!), it soon becomes necessary to define a standard numerical representation of length that can be used to measure the length of any arbitrary object. Modern measurement systems for physical attributes such as length, mass, time, etc. provide such universal scales.

To give another example, let us return to the autonomous vehicle application, and consider the evaluation of several alternative braking systems for a vehicle. Since the primary purpose of a braking system is to stop the vehicle, we might choose to measure the utilities of the braking systems in terms of their stopping distances under specific road conditions, and for a specific speed of travel (for example, say, the stopping distance from 55km/h on a good, dry road). If the set of alternative braking systems is $A = \{A_1, A_2, A_3\}$, then we can record our empirical observations of how well the alternatives perform as follows:

$$\begin{aligned}\mathcal{R} = \{\ &(A_2 \text{ has more utility than } A_1),\\ &(A_3 \text{ has more utility than } A_1),\\ &(A_2 \text{ has more utility than } A_3)\}\end{aligned}$$

where "has more utility than" means "has a smaller stopping distance than".

If $\mathbf{N}$ is chosen to be the set of integers, then we could define a homomorphic mapping as follows:

$$\mathbf{M} = (\{(A_1 \mapsto 1), (A_2 \mapsto 3), (A_3 \mapsto 2)\}, \mathcal{R} \mapsto >)$$

Once more, an infinite number of possible homomorphisms exist. Moreover, the empirical relations might be different for braking in the wet, and for braking from different speeds. In order to make a complete evaluation of the alternatives, it may be necessary to construct measures of utility over a range of benchmark conditions. These multiple measures may then need to be combined using a 'multiple-attribute utility function', but this is beyond the scope of this paper.

#### 4.1.2. *Scale Types*

If $\mathbf{M}$ is a representative mapping from $\mathbf{E}$ to $\mathbf{N}$, then $(\mathbf{E}, \mathbf{N}, \mathbf{M})$ is called a *scale*. There are three important questions concerning representations and scales:

1. How to choose one $\mathbf{N}$ in preference to another?
2. How do we know if a particular $\mathbf{E}$ has a representation in a given $\mathbf{N}$?
3. What do we do when we have a number of different possible homomorphisms to the same $\mathbf{N}$?

The answer to the first question is pragmatic (choose the real number system $\Re$ where possible, else find a simpler system), the second question is essentially the *representation problem* and the third is the *uniqueness problem*. The representation problem is considered to be one of the basic problems of measurement theory. It has been solved in different disciplines for various types of relation systems characterised by certain types of axioms. Krantz *et al.* in their seminal book (Krantz et al., 1971) have collected together a number of theorems from various sources and attempted to axiomatise them, i.e., show that most of them derive from a small set of basic theorems which serve as axioms of the representational theory of measurement.

The uniqueness problem is also related to the notion of scales. Although there may be many different possible representations for the same empirical relation system, any two representations may be related in a very simple way. If $\mathbf{M}$ and $\mathbf{M}'$ are two acceptable representations (i.e. homomorphisms) of the same empirical relation system, each is called a *re-scaling* of the other, via a mapping called an *admissible transformation*. In other words, given an $\mathbf{M}$, an $\mathbf{M}'$ can always be derived from it by an admissible transformation, and conversely, given an $\mathbf{M}$, if we apply an admissible transformation on it, we arrive at an $\mathbf{M}'$ which is an acceptable representation. An example here would be the tranformations between $^oC$ and $^oF$ for measuring temperature.

The class of admissible transformations for a given empirical relation system defines how unique the scale is. It defines the *scale type*. In other words, the scale type for a given empirical relation system is determined by the rescalings which are allowed. The fewer possible acceptable rescalings for a given relation system, the more restrictive will be the scale type. We will continue discussion of this topic in Section 4.3.

Returning to the vehicle braking example, we could consider enriching the empirical relation system by translating observations of the following type :-

$A_2$ stops in 20 feet, but $A_1$ stops in 10 feet,

which we may, or may not, wish to interpret as $A_2$ is *twice as good as* $A_1$, and

$A_3$ stops in 30 feet, but $A_1$ stops in 20 feet,

could mean that $A_3$ is 1.5 times better than $A_1$, etc. The measurement mappings required to represent such observations would result in stronger scales than the mapping $\mathbf{M}$ defined earlier.

### 4.2. Formalising 'Value' using Measurement Theory

Suppose $A$ is the set of alternatives (or services) currently under consideration and the binary relation

$$A_i \text{ Is More Useful Than } A_j$$

holds if, and only if, alternative $A_i$ is judged to have more utility than $A_j$. Then we would like to assign a real number $V(A_k)$ to each $A_k \in A$ such that for all $A_i, A_j \in A$,

$$(A_i \text{ Is More Useful Than } A_j) \Leftrightarrow (V(A_i) > V(A_j)) \tag{3}$$

The function $V$ is called an ordinal utility function or ordinal value function, and $V(A_k)$ is the utility value of $A_k$. If we can find such a value function, we can use it to make decisions: for example, always choosing the alternative(s) with the highest value(s).

We may also demand more of the value function: we may want it to be additive in the sense that the value of running two services is the sum of their values. Formally, we need a binary operation $\circ$ for combining elements in the set $A$, where the meaning of $(A_i \circ A_j)$ must be defined as required by the application. Whatever the interpretation of $\circ$, an additive value function is a real-valued function $V$ on $A$ which not only satisfies condition Equation (3), but also preserves the binary operation $\circ$, in the sense that for all $A_i, A_j \in A$,

$$V(A_i \circ A_j) = V(A_i) + V(A_j)$$

Returning to the example of length measurement, we could define the operator $\circ$ to mean "placed end to end with", so that we could calculate the combined lengths of two or more objects. On the other hand, there is no such obvious concatenation operator for combining braking systems to calculate their combined utilities. However, it may be possible, and useful, to define a concatenation operator for combining the utilities of other services and alternatives. Note that many of these concatenation operators will not be additive, for example, although the utility of running two collision avoidance service alternatives would be more than that of running only one, it is unlikely to be as much as the sum of their individual utilities. However, the utility of running one service followed by (or in parallel with) a different service, might well be the sum of their utilities.

### 4.3. Meaningfulness: The Link between Assignment and Use of Value

The theory of scale types can be used to describe the notion of meaningfulness. This is the formal mechanism within which one can reason whether or not statements about measurement 'make sense'. This also tells us about the limitations on the kind of mathematical manipulations that can be performed on the numbers arising as scale values. Although we can always perform operations such as adding, averaging, and taking logarithms, we need this formal notion to know if we can still deduce meaningful statements about the entities in terms of these operations.

Measurement theory accounts for meaningfulness by considering the uniqueness of the homomorphisms satisfying the representation condition. Since there may be many different homomorphisms from an empirical relation system into a numerical relation system, the only meaningful statements are those which specify which scale (homomorphism) is being used, or, those which make sense independent of the scale. This is formalised as:

> A statement involving numerical scales is *meaningful*, if its truth (or falsity) remains unchanged if every scale $\mathbf{M}$ is replaced by an acceptable rescaling $\mathbf{M}'$.

Meaningfulness is distinct from the notion of truth: there may be statements which are perfectly meaningful, yet clearly untrue. For example, the statement "An empty car weighs more than the same car filled with a load" is clearly false. However the statement is meaningful because its falsity remains unchanged no matter which scale of measurement of weight is used for the comparison.

Table III shows the most commonly used scales of measurement, their admissible transformations, and examples of attributes measurable on those scales (as characterised by specific empirical relation systems). The table also shows examples of meaningful statements involving values that can be used on different scale types. These examples are representative of the 'strongest' arithmetic allowed on each scale as determined by the list of comparison operators and functions that are meaningful when applied to values on that scale. The rationale for the validity of the examples is provided by the admissible transformations column.

*Thus there is an important theoretical link between the method of generating the numbers representing utility values, and the use of these numbers in particular scheduling or admission strategies. This link is poorly recognised in the scheduling literature. For any value-based scheduling technique to be sound, it must take into account the suitability of the scale upon which the values are measured.*

Note that the scale types in Table III are ordered. The scale type in each row implies all the properties of the earlier scale types. For example, a ratio scale is also ordinal. A utility function which is stronger than ordinal is often called a *cardinal* utility function; i.e. it is either interval, ratio or absolute.

As we will discuss below, the more 'sophisticated' the operations performed on value, the stronger the scale upon which it must be measured, and hence, the richer the underlying empirical relation system must be. This in turn implies that our knowledge and understanding of the relative utilities of the services must fulfil the conditions necessary to map the empirical relation system into the required scale. That is, they must satisfy the associated representation theorem. This knowledge

Table III. Scales of Measurement and Examples of Valid Uses of $V$

| **Scale Type** | **Admissible Transforms** | **Everyday Examples** | **Meaningful Comparison Operators** | **Meaningful Operators, Functions** | **Examples of Valid Use of $V$** |
|---|---|---|---|---|---|
| Nominal | $\mathbf{M}' = F(\mathbf{M})$, $F$ is 1 to 1 | Labeling, Classification | $=$ | mode | $V_1 = V_2$ |
| Ordinal | $\mathbf{M}' = F(\mathbf{M})$, $F$ monotonic increasing | Hardness | $=, <, >,$ | mode, median | $V_1 \leq V_2$, $V_1/k \geq V_2/k$ |
| Interval | $\mathbf{M}' = \alpha\mathbf{M} + \beta$, $\alpha, \beta > 0$ | Temperature ($^oC/^oF$), Calendar time, Position | $=, <, >,$ distance | mode, median, mean, minus | $V \geq V_{mean}$, $V_1 - V_x \leq V_2 - V_y$ |
| Ratio | $\mathbf{M}' = \alpha\mathbf{M}$, $\alpha > 0$ | Length, Temperature (Kelvin), Time interval | $=, <, >,$ distance, ratio | mode, median, mean, minus, plus, $*$, $/$, $\%$ | $V_1/C_1 \leq V_2/C_2$, $V_1 \geq 20\% V_2$ |
| Absolute | $\mathbf{M}' = \mathbf{M}$ | Counting | All | All | $V = 10$ |

comprises a mixture of engineering data, and engineering judgements provided by domain experts. The larger the contribution of engineering data, the more likely we are to be able to achieve richer scales of measurement (interval or ratio).

We discuss the relevance of the scale type of value to particular scheduling algorithms in more detail below.

#### 4.3.1. *Direct Comparison of Values*

Values on the nominal scale are only useful as a means of classification, with no implied ordering between them, even if the numbers appear to be ordered. For example, two alternative breaking algorithms $A_1$, and $A_2$ could be assigned the values 20 and 30, respectively, without implying any ordering between them. The only conclusion the scheduler can draw is that their values are unequal. This is analogous to the implementation of enumerated types in programming languages.

If the scheduler needs to pick out the highest valued alternative to execute, then the values must be assigned on at least an ordinal scale. Thus if the values assigned to $A_1$, and $A_2$ are 20 and 30, on an ordinal scale, then the scheduler can conclude that $A_2$ has more utility than $A_1$. Of course, any monotonic-increasing function (e.g., multiplication or division by a constant) can be applied to the values uniformly without changing their relative ordering. However, the scheduler cannot infer, for example, that $A_2$ is 1.5 times more useful than $A_1$ (since the scale is not ratio), or even that $A_2$ is 10 units more useful than $A_1$ (since the scale is not even interval).

### 4.3.2. *Use of 'Average' Values*

The meaningfulness of the notion of average also depends on the scale type. As shown in Table III, the only kind of averaging operation which can be applied to values from a nominal scale is that of mode, but for ordinal values their median is also meaningful. However, the arithmetic mean of a set of values, cannot be calculated unless they are on at least an interval scale[2].

Thus, for example, an admission policy which only admits services whose values are higher than a mean value threshold, is only valid if the values were assigned on an interval (or higher) scale. The representation condition for an interval scale requires that we should be able to construct empirical relations such as "the difference between the utility values of services $A_1$ and $A_2$ is bigger than the difference between the uility values of services $A_3$ and $A_4$". Empirical observations of this form would then be used to derive the numerical relation system, which provides the utility values to be fed into the scheduling policy. For any particular application, if the domain knowledge is too poor to make such observations about the relative utilities of the services, then the theory says that the knowledge cannot support an interval scale. Hence the calculation of arithmetic means, and the use of mean value threshold policies for scheduling, would be meaningless for the application.

Papers in the value-based scheduling literature that advocate a mean value threshold approach have not recognised these constraints on the derivation of values.

### 4.3.3. *More Sophisticated Uses of Value*

With stronger scales of measurement it is possible to use values in more interesting ways. For example, the popular value density scheme (see Section 3.2) becomes meaningful on the ratio scale, but is infeasible even on the interval scale. It is instructive to demonstrate this through a worked example.

Consider the proposition that $V_1/C_1 > V_2/C_2$ is true, where $V_1$ and $V_2$ are measured on an interval scale. Let $V_1'$ and $V_2'$ be the results of applying admissible transformations on $V_1$ and $V_2$ respectively. Then we should be able to show that the truth of the above proposition is unaltered when the measures $V_1$ and $V_2$ are substituted by $V_1'$ and $V_2'$ respectively. In other words, we need to show that:

$$\begin{array}{lllll} & V_1'/C_1 & > & V_2'/C_2 & \\ \Rightarrow & (\alpha V_1 + \beta)/C_1 & > & (\alpha V_2 + \beta)/C_2 & \textit{(substitute definitions of } V_i'\textit{s)} \\ \Rightarrow & \alpha V_1/C_1 + \beta/C_1 & > & \alpha V_2/C_2 + \beta/C_2 & \textit{(expand)} \\ \Rightarrow & \alpha(V_1/C_1 - V_2/C_2) & > & \beta(1/C_1 - 1/C_2) & \textit{(collect like terms)} \end{array}$$

We know that the left hand side of the last inequality above is positive (since $\alpha$ is is positive and $V_1/C_1 > V_2/C_2$ is true). However, the right hand side could be either positive or negative, depending on the values of $C_1$, $C_2$ and $\beta$, and when it is positive, it might be greater than or equal to the left hand side. Hence, the truth of the last inequality is not guaranteed by the truth of the original proposition. This is why the original proposition is meaningless, at least for interval scale values of $V_1$ and $V_2$.

Now consider the case that $V_1$ and $V_2$ are measured on a ratio scale. In this case, the admissibly transformed measures $V_1'$ and $V_2'$, when substituted in the original proposition, give rise to the inequality:

[2] Note that the arithmetic mean of a set of numbers is a valid property on the interval scale, even though the plus operator is not meaningful until the ratio scale. This is because the mean can be defined in terms of the differences between values (i.e. using the minus operator, which is available on the interval scale!).

$$
\begin{array}{llcll}
 & V_1'/C_1 & > & V_2'/C_2 & \\
\Rightarrow & \alpha V_1/C_1 & > & \alpha V_2/C_2 & \textit{(substitute definitions of } V_i'\textit{s)} \\
\Rightarrow & V_1/C_1 & > & V_2/C_2 & \textit{(cancel } \alpha\textit{)}
\end{array}
$$

The truth (or falsity) of the last inequality is always guaranteed by the truth (or falsity) of the original proposition. Hence the use of $V$ in an admission policy based on $V/C$ is meaningful on the ratio scale, but not on any weaker scale than the ratio scale. Scheduling literature that advocates the use of schemes such as $V/C$, or $V/D$, has not noted this requirement on $V$.

Returning to the example of the use of $V/C$ in Table II, we note that the calculation of value-density, $\Omega$, would be invalid unless the numbers for $V$ were generated on (at least) a ratio scale. In fact, the technique which was used in this case (Saaty, 1992), is known to generate values on a ratio scale by assuming the availability of sufficient data to support the richness of the empirical relation system required by a ratio scale. For an application such as the autonomous vehicle example, this amounts to having sufficient domain expertise and engineering data to make claims such as "the service that provides ABS braking is approximately 4.5 times more important than the service that calculates and displays speed warnings".

#### 4.3.4. *Use of Time-Varying Value Functions*

It is common in the scheduling literature to define a value-function that varies with time. Some definitions assume that the value remains constant until the service's deadline, and then either falls to zero immediately, or decays to zero over a further period of time. Other definitions might allow the value to vary before the deadline, in order to represent varying degrees of hard and soft deadlines.

Whatever the scale type of the value function (ranging from nominal to absolute), the interpretation and use of the function's result must be consistent with its scale, at every point in time. Furthermore, if the values of a service at different times are to be compared, then this comparison should also be done in accordance with the scale type. For example, the concept of having 'half the value after $t$ milli-seconds' cannot be used unless the scale of the value function is at least ratio. Although this is a commonly used scheduling approach for trading value delivered against time, this requirement on the scale type of the value function has again not been emphasised.

#### 4.3.5. *Use of Value in Simulation Studies*

In almost all papers on value-based scheduling, evidence of effectiveness is derived via simulation studies. The typical metric used is *value-sum*. That is, for all completed services or alternatives, the values accrued are totalled and this is used to compare one scheduling algorithm with another. However, this metric is only valid if the scale on which the values of the services are defined is no weaker than the ratio scale, because the plus operator is meaningless on weaker scales. If a simulation study uses average values rather than value-sum, then an interval scale suffices, as discussed in Section 4.3.2.

## 5. Conclusions

Real-time systems requiring autonomous control involve scheduling strategies that use the concept of utility value to achieve their adaptive run-time behaviour. This paper addresses a problem that was hitherto poorly recognised in the literature of scheduling: the ad hoc numbers representing utility values were not being checked for conformance to the scales of measurement implied by the scheduling schemes.

The paper has discussed the measurement of values on well-understood scales (nominal, ordinal, interval, ratio and absolute), and shown how to relate the assumptions underlying these scales to the usage of the values in various scheduling policies. The scale type determines the operations allowed on values of the type and hence the form of scheduling algorithm that is meaningful. It is important that a stronger scale is not assumed during scheduling than has been obtained during assignment.

Although most papers on dynamic scheduling assume a rich scale for value, it is doubtful whether such richness will actually be furnished by the applications of interest. More scheduling results on weaker scales are required for use in these situations.

## References


AEEC: 1991, 'ARINC 651: Design Guidance for Integrated Mondular Avionics'. Airlines Electronic Engineering Committee, draft edition.

Biyabani, S., J. Stankovic, and K. Ramamritham: 1988, 'The Integration of Criticalness and Deadlines in Scheduling Real-Time Tasks'. In: *Proceedings of the 9th IEEE Real-Time Systems Symposium*. pp. 152–169.

Burns, A. and D. Prasad: 1998, 'Value-Based Scheduling of Flexible Real-Time Systems for Intelligent Autonomous Vehicle Control'. In: M. Salichs and A. Halme (eds.): *Proceedings of the 3rd IFAC Symposium on Intelligent Autonomous Vehicles*. Madrid, Spain, pp. 127–132.

Burns, A., D. Prasad, A. Bondavalli, F. D. Giandomenico, K. Ramamritham, J. Stankovic, and L. Strigini: 2000, 'The Meaning and Role of Value in Scheduling Flexible Real-Time Systems'. *Journal of Systems Architecture*.

Burns, A. and A. Wellings: 2001, *Real-Time Systems and Programming Languages*. Addison Wesley, 3rd edition.

Buttazzo, G. and J. Stankovic: 1993, 'RED: A Robust Earliest Deadline Scheduling Algorithm'. In: *Proceedings of the 3rd International Workshop on Responsive Computer Systems, Austin*.

Davis, R.: 1995, 'On Exploiting Spare Capacity in Hard Real-Time Systems'. Ph.D. thesis, University of York, Department of Computer Science. D.Phil. Thesis.

Davis, R., S. Punnekkat, N. Audsley, and A. Burns: 1995, 'Flexible Scheduling for Adaptable Real-Time Systems'. In: *IEEE Real-Time Technology and Applications Symposium*. pp. 230–239.

Fenton, N.: 1991, *Software Metrics: A Rigorous Approach*. Chapman and Hall.

Huang, J., J. Stankovic, D. Towsley, and K. Ramamritham: 1989, 'Experimental Evaluation of Real-Time Transaction Processing'. In: *Proceedings of the 10th IEEE Real-Time Systems Symposium*.

Jensen, E., C. Locke, and H. Tokuda: 1985, 'A Time Driven Scheduling Model for Real-Time Operating Systems'. In: *Proceedings IEEE Real-Time Sytems Symposium*. pp. 112–122.

Krantz, D., R. Luce, P. Suppes, and A. Tversky: 1971, *Foundations of Measurement: Additive and Polynomial Representations*, Vol. 1. New York: Academic Press.

Locke, C.: 1986, 'Best Effort Decision Making for Real-Time Scheduling'. Ph.D. thesis, Computer Science Department, CMU. PhD Thesis, CMU-CS-86-13.

Prasad, D.: 1998, 'Dependable Systems Integration Using Measurement Theory and Decision Analysis'. Ph.D. thesis, Department of Computer Science, University of York, U.K.

Prasad, D. and A. Burns: 2000, 'A Value-based Scheduling Approach for Real-Time Autonomous Vehicle Control'. *Robotica* **18**.

Roberts, F.: 1979, *Measurement theory, with applications to decision-making, utility and the social sciences*. Reading, MA: Addison Wesley.

Saaty, T.: 1992, *Multiple Criteria Decision Making: The Analytic Hierarchy Process*. Pittsburgh, PA: RWS Publications.

Tokuda, H., J. Wendorf, and H. Wang: 1987, 'Implementation of a Time-driven Scheduler for Real-Time Operating Systems'. *Proceedings of IEEE Real-Time Systems Symposium* pp. 271–280.

Wendorf, J.: 1988, 'Implementation and Evaluation of a Time-driven Scheduling Processor'. *Proceedings of IEEE Real-Time Systems Symposium* pp. 172–180.


*Address for Offprints:*
Please send offprint requests to **divya@mca-ltd.com**