Let’s give Availability back to the engineers

Availability (with a capital ‘A’) is a well-defined and long established engineering parameter. It even has a British Standard definition relating to the percentage of time that a system is available (with a small ‘a’) to be used i.e. is not faulty. With the advent of modern digital systems, new issues arise that can prevent a system from being used for reasons other than that they are faulty and the term Availability has been hijacked by the Service Level Agreement (SLA) industry to include these periods. This is leading to extremely muddled thinking because Availability, used in this way, combines at least three different, independent, mechanisms and this prevents any attempt at rational analysis or mathematical description. Worse, it muddles the responsibilities for corrective action and perpetuates a sloppy approach to engineering design.

Intercai finds this unprofessional approach deplorable and in this short paper explores the different mechanisms and proposes an approach that does offer the ability to analyse what is going on and describe the mechanisms mathematically. We believe that the term Availability should revert to its classical meaning and, for the avoidance of doubt, we propose a supplementary definition that states that Availability is the proportion of time during which a system operates as its designers intended (i.e. it is not faulty and needs no repair).

We now need to introduce a new term to cover periods when a system is Available (in the sense defined above) but still does not provide the expected service to the user, for whatever reason. We propose to introduce the term Usable. A system is Usable when it is delivering the service, at the levels of performance prescribed, that the user expects. The parameter that measures this characteristic is Usability. Usability is a derived parameter that combines at least three mechanisms that can cause a system to become unusable.

The first of these is the classic Availability. When a system is faulty, broken, or in any way different to the designer’s intent and requires corrective action – then it is unavailable. Classically, this is a fault, and the process of restoration is a repair. Both mechanisms are susceptible to normal engineering analysis and specification and the responsibility for corrective action rests with the designers or operators of the system.

The second cause of an unusable system occurs when it is running exactly as the designers intended, but is subjected to more traffic or load than it was designed for (the M25 effect).

Performance degrades below the specified level, or stops altogether, and the system is not usable. In this instance, the designer or operator cannot correct the problem, because the system is not faulty. Obviously more capacity can be added, but who should bear the cost of this? The overload may be temporary, or due to some one-off event, and the user may be unwilling to pay for enhancements that he does not need. This mechanism is entirely different from that of the failure described above. The cause of the problem lies entirely with the collective group of users (who can force the condition at will). It is susceptible to engineering analysis, although the underlying mathematics is completely independent of the failure scenario. The solution, though, must be worked out between the operator and the user on the basis of what makes best business sense.

There is a third mechanism that is becoming accepted as a normal event as systems become more complex. Some suppliers of management software address this as the main issue in network management and make a primary selling point out of its existence, by offering tools to detect and diagnose its existence. This mechanism is the mis-configuration, or operator, error when some person with access to the system takes some control action that interferes with its operation with a consequent effect on usability. This mechanism is not susceptible to mathematical analysis in the same way as the other scenarios described above. It arises as a result of inadequate training, insufficient preparation for the change or incomplete testing of the change or the processes that support it. Clearly it is the responsibility of the operator to correct this event, but we take the view that it is entirely avoidable and should not occur in the first place. This is one category of events that should be punished with the full might of the SLA mechanism. If a system is so complex that a level of operator mis-configuration has to be accepted as normal, then it is our view that such a system is unsafe and should not be introduced into service.

In the current climate of retrenchment, we see the power moving rapidly back from the equipment suppliers to the operators. We sincerely hope that the operators make use of this opportunity to insist that suppliers stop delivering new functionality at the expense of adequate testing, and that they make use of the ample capacity that has been installed to simplify their architectures and management systems as the unit prices fall. They have the chance to make sure that changes are not applied to a system by poorly trained staff, or without adequate pre-testing. Change procedures should ensure that if there is a consequential effect, in spite of all the precautions, there is the ability to return to the *status quo* to enable the system to continue running while the reasons for the effect are established. Now is the time to reduce the incidence of operator error as a cause of system unusability to an insignificant level (and incidentally delete a whole layer of administrative software aimed at detecting and diagnosing such events).

Let us give Availability back to the engineers and if we must have a catch-all parameter, then introduce ‘Usable’, which itself depends on Availability, Load v Capacity and (if we must) operator error. We must remember that these three mechanisms are entirely unrelated and independent and any one of them could swamp the others in terms of delivered performance. Let us also stamp on the sloppy approach to engineering design that encourages us to accept ‘operator error’ as a fact of life that has to be lived with, rather than an entirely avoidable consequence of over-complexity and inadequate preparation for entry into service.

steve.hodson@intercai.co.uk **Tel: +44(0) 1628 478 470**
Intercai Mondiale Ltd, Regatta House, High Street, Marlow, Bucks SL7 1AB, UK
www.intercai.co.uk