

Dealing with Availability in an international Service Management scenario

Flavio Gaj¹, Giovanni Umberto Germani²

¹ Head of ICT Operating Processes & Quality of UniCredit S.p.A, Milan, Italy

² Department of Computer Engineering and Systems Science, University of Pavia, Italy

Abstract: In today's global economy context, since companies rely heavily on their information systems to delivery services to their Customers, any downtime can have serious impacts on their business. At the same time, the traditional approach of measuring the availability of only systems and components is no longer sufficient and must give way to a more comprehensive measurement that takes account of the whole service delivery chain. In this paper we present the method used by an international ICT service provider as UniCredit Group, to monitor and improve the availability of its services.

1 Introduction

In today's global economy context, organizations of every kind and in all parts of the world have become increasingly dependent on their IT systems. E-commerce made it possible to conduct business 24 hours a day, 7 days a week. New, powerful applications allow businesses and institutions to introduce unprecedented levels of computerization into their daily operations, to the point that nowadays we all depend on reliable access to computer systems at all times.

This need has dramatically increased the importance of systems availability. Since companies and organizations rely heavily on computer systems to conduct their business, any downtime can seriously cripple their business. More than just lost productivity, downtime has come to mean lost revenues and even weakened market position. At the very least, IT downtime can severely impact a business' operations and increase cost enormously.

Availability means that a system is on-line and ready for access [1]. A variety of factors can take a system off-line, ranging from planned downtime for maintenance to catastrophic failure. The goals of high availability solutions are to minimize this downtime and/or to minimize the time needed to recover from an outage. Exactly how much downtime can be tolerated will dictate the comprehensiveness, complexity and cost of the solution.

High availability is a convenient label, but its meaning is often misunderstood. High availability is not a specific technology nor a quantifiable attribute. Rather, it

Please use the following format when citing this chapter:

Gaj, F. and Germani, G.U., 2008, in IFIP International Federation for Information Processing, Volume 280; *E-Government; ICT Professionalism and Competences; Service Science*; Antonino Mazzeo, Roberto Bellini, Gianmario Motta; (Boston: Springer), pp. 235–244.

is a goal to be reached, one that has different meanings according to need. A variety of strategies, technologies and services are used to accomplish that goal.

Following the new trend which sees the transition to a Service Oriented Economy [2], the simple concept of availability of IT systems (the accessibility of a system resource in a timely manner, i.e. the measurement of a system's uptime) is no longer sufficient and should be extended to a wider concept of *Services Availability*.

2. Approaching Service Availability

Let UT be the Up-Time, i.e. the time interval in which the service¹ is running in the desired Operating Window OW , then in its most elementary form, service availability A is defined as:

$$A = \frac{UT}{OW} \times 100 \quad (1)$$

In assessing service availability in a real, complex ICT scenario, the designer has to find the right balance between:

- Costs of measures.
- The significance of the parameters.
- The effectiveness of the measure as a mean of communication between Customers and Supplier, and as an analysis tool of downtimes aimed at improving the service quality.

A first element of complexity is due to the fact that sometimes an incident may impact only a portion of a service, causing the temporary unavailability of some features, while others remain usable by the Customer. In these circumstances, the designer must face one of the following alternatives:

1. Define as a “Service” each atomic set of functionalities, and measure the availability for each of them (at the most detailed level, the single physical transaction).
2. Define as a “Service” a comprehensive set of functionalities, and weigh the downtime of a limited portion of functionalities with a corrective factor ranging from 0 to 1.

The pros and cons of the two solutions are self-explanatory. In the first case the associated costs may be high, but the information is so detailed and analytic that no misunderstanding may arise. In the second case the costs are lower, (the indicator is only one), but the algorithm may lead to conclusions that someone

¹ In defining what a Service is, we embraced an “end-to-end approach”, looking at services as Customers do. [3]

may consider to be arbitrary. Our method follows the middle road between these two approaches, as explained in next section.

3 Measuring Service Availability in a Complex Context

In measuring service availability, we use a granularity which takes into consideration a comprehensive and exhaustive set of functionalities, without descending to the level of detail of single transactions. Furthermore, we choose a set of functionalities that has the ability to be significant for the main counterpart (the Customer).

In our model each Service is split into the relative SubServices, in turn divided into a set of Components. A weight is assigned to each SubService/Component, modeling its importance compared to the Service/SubService to which it belongs.

Table 1. An example of Service breakdown structure

SERVICE	SUBSERVICE	SUBSERVICE COMPONENT	COMPONENT WEIGHT vs SUBSERVICE	SUBSERVICE WEIGHT vs SERVICE	
Current Accounts and Deposit	Deposit and Current Accounts		100	20	
		Cash pooling	5		
		Conditions	15		
		Current Accounts	40		
		Customer documents	5		
		Printing of statements and communication to the customer	15		
		Saving Accounts	10		
			10		
	Other component			100	20
		Archive of Teller's journal	40		
		Suspended items	60		
	Safe boxes			100	20
		Safe Boxes	100		
	Transfer Payments, Collection			100	20
		Cheques	100		
	Corona			100	20
Nostro/Loro Accounts Reconciliation		100			

When an incident occurs, the following data are collected:

- Start/End time (*Duration*)
- Incident cause (code/sub-code, for eventual analysis)
- Incident Description
- Service/SubService/SubService Component impacted
- Weight for each SubService Component

Let i be the number of incidents happened into the Operating Window OW , then the daily availability A is:

$$A = \frac{OW - \sum_1^i Duration_i \times SSWeight_i \times SSCWeight_i}{OW} \tag{2}$$

where $SSWeight$ represents the weight of the SubService vs Service and $SSCWeight$ is the weight of the SubService Component vs SubService.

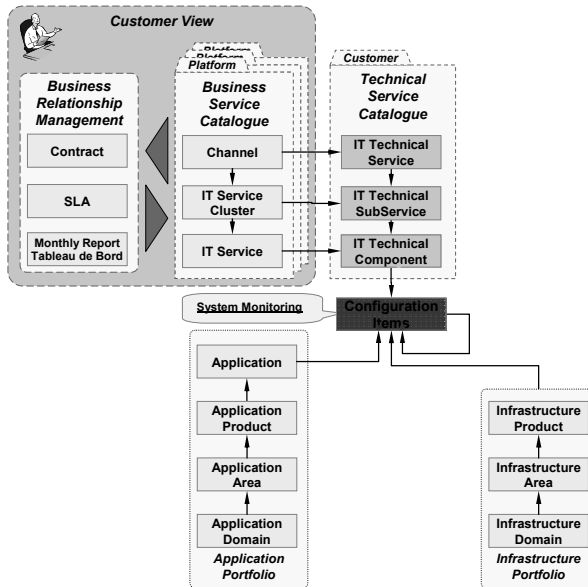


Fig. 1. Service monitoring through Configuration Items

Our approach provides a continuous monitoring of the service status through the use of Configuration Items. A Configuration Item (CI) is a component (IT Infrastructure element) that can be handled individually (i.e. computers, routers, servers, software...) [3]; furthermore, for each component we consider two reliability parameters, well known in the manufacturing industry and literature: MTBF and $MTTR^2$ [4].

A service is up and running if a number of CIs are up and running together and simultaneously. The number of CIs for complex services may be very high (20-30

² For the purpose of this document we define MTTR as the whole time needed to manage the incident, including factors that do not belong to the classic definition of MTTR, such as: the way in which the fault is detected, the contractual agreement with the servicing contractor; furthermore we consider any eventual set-up time necessary after the restoring of the device (T_{setup}).

elements, depending on the level of details of the analysis). We will define this ensemble of components *Service Chain*. The service chain has to be entirely up and running to be able to deliver the whole service.

When dealing with service availability, a common objection is that the problem does not exist because a proper architecture is redundant, and a fault results in the takeover of the redundant component without service downtime.

This is generally true. It is also true; however, that in our experience the occurrence of hardware faults has a very low probability. When they seldom happen, if the redundancy works properly and the switching mechanisms are efficient, the faulty component is isolated and the operation goes on based on the redundant component. They are formally recorded for statistical reasons, but the down-time is set to zero, as they don't impact the continuity of the service. Sometimes the eventual operation to restore the faulty component may jeopardize the Service, although these operations are scheduled (whenever possible) outside the operating window. If a downtime is required or caused, the event is recorded as a fault.

On the other side, software and microcode faults have a dramatically higher probability of occurrence. In these cases obviously the redundancy does not work. The software/firmware controls virtually every functionality at the most detailed level of IT devices, with millions and millions of Lines of Code (LOCs) in any business environment of medium or even low complexity. From a purely theoretical point of view, software errors should decrease over time, because an error, once happened and correctly fixed, does not happen again: this is true. Nevertheless, IT industry is very dynamic and the never-ending innovation and increasing competition push vendors to launch on the market new products, new versions and new releases in more and more smaller time intervals. These bring to the users new functionalities and opportunities, but also new bugs. It is the common opinion that this trend will increase in the next coming years. [5]

For these reasons, it is reasonable, although not fully theoretically correct, to use MTBF and MTTR as parameters representing software reliability. Based on the above assumptions, the theoretic availability A_{Tring} of a single ring of the service chain is:

$$A_{Tring} = \left(1 - \frac{MTTR_{\min} + T_{\text{setup}_{\min}}}{MTBF_{\text{year}} \times 365 \times 24 \times 60} \right) \times 100 \quad (3)$$

The formula is related to operating window "24x7", but in reality the *OW* may be narrower and we may face different situations. Let's define A_{Emx} as the resulting effective maximum availability in different hypothesis.

1. The fault probability is fully random vs operating window. In this case the downtime is proportional to operating window.
2. The faults happen only in the operating window. The whole downtime calculated for the window 24x7 falls in a shorter window, so decreasing the resulting availability.

We experienced that some faults are really random (e.g. power supply failure), while other are for sure depending on the fact that the devices are really working. These second type of faults more likely fall in the operating window. Let $ATmx$ be the theoretical maximum availability of the service chain considering faults as random events, then the effective maximum availability may be calculated by following formula:

$$AEmx = \frac{OW_{year} - ((1 - ATmx) \times 365 \times 24 \times 60) \times Percentage_{OW}}{OW_{year}} \quad (4)$$

where $Percentage_{OW}$ represents the estimation of the percentage of the faults assumed to fall within the operating window. The table below shows the resulting availability in some hypothesis of percentage of faults falling into operating window, in some of the more significant operating windows (we consider the service chain of a typical Web application, with an estimated $ATmx$ of 99,1914%). The percentage of faults in window is assumed to be higher as the window is wider. If we assume, for example, 100% in window for Internet Services (20x7), and 40% for Transactions Driven Applications (day window), the expected availability would be round 99%.

This was an example of a very long and complex service chain. Simplifying it and reducing the number of rings can reduce significantly the down times due to infrastructures. Some trends in infrastructure help in reducing the number of rings: for example virtualization allows basing a number of instances on the same processor. Assuming that the MTBF of a more powerful processor complex is the same of smaller processor complex, we reduce the probability to have a hardware failure in the time frame under consideration.

Table 2. Service availability with different $Percentage_{OW}$

Operating Window			Year Window [m]	Percentage of faults in OW					
h/d	d/w	w/y		0%	20%	40%	60%	90%	100%
1	2	3	4	0%	20%	40%	60%	90%	100%
11	5	52	171.600	100,00%	99,50%	99,01%	98,51%	97,77%	97,52%
20	7	52	436.800	100,00%	99,81%	99,61%	99,42%	99,12%	99,03%
14,5	5	52	226.200	100,00%	99,62%	99,25%	98,87%	98,31%	98,12%
13	5	52	202.800	100,00%	99,58%	99,16%	98,74%	98,11%	97,90%
18	5	52	280.800	100,00%	99,70%	99,39%	99,09%	98,64%	98,49%
13	5	52	202.800	100,00%	99,58%	99,16%	98,74%	98,11%	97,90%
10	5	52	156.000	100,00%	99,46%	98,91%	98,37%	97,55%	97,28%
12	5	52	187.200	100,00%	99,55%	99,09%	98,64%	97,96%	97,73%
24	7	52	524.160	100,00%	99,84%	99,68%	99,51%	99,27%	99,19%

The downtimes due to application software and operational errors can be calculated on a statistical basis as well, based on the expected number of modifications to running systems. Application software and operation error are man-made mistakes; the incidence of mistakes is higher as high is the change rate, both on the application side and the technology side. The fault reasons listed above must be added to the downtimes estimated by the model.

The impact of application software maintenance in a fast changing business environment may be very high. Short time-to-market can be enemy of stability and availability: exhaustive tests should be balanced with time-to-market case by case.

Technology maintenance is another important source of errors. The number of errors is assumed to be a function of the change rate.

The most important areas to be considered are the following:

Hardware maintenance. Often it's necessary to agree windows of downtime to change/upgrade CPUs, disks control units, or to substitute redundant faulty components that can be substituted at a later time (if possible outside the operating windows, but for sure within the window for 24*365 services).

Software microcode maintenance. In many cases the upgrade of microcode and system software requires the stopping of the devices, with a consequent downtime. For services running "round-the-clock" in these cases the daily availability may severely reduced.

Application software maintenance. In same cases the upgrade of business functionalities or the release of new versions compel the stopping of operations for data migration or rebuilding, and sometimes it may be impossible to do that outside the operating window.

4. UniCredit Group Case Study

Ranking among the top financial groups in Europe, UniCredit has banking operations in 23 countries and international network spanning 50 countries, with about 10.000 branches and approximately 180,000 employees at 31 march 2008. Besides Italy, Germany and Austria, UniCredit operates - in the CEE region - the largest international banking network with about 4.000 branches and outlets and about 80,000 employees. The Group operates in the following countries: Azerbaijan, Bosnia and Herzegovina, Bulgaria, Croatia, the Czech Republic, Estonia, Hungary, Latvia, Lithuania, Kazakhstan, Kyrgyzstan, Poland, Romania, Russia, Serbia, Slovakia, Slovenia, Tajikistan, Turkey and Ukraine. The studies and projects regarding availability (both on cultural side and measurement algorithms and tools) started in 2002, and allowed UniCredit Group to manage effectively its very complex international ICT service providing structure.

UniCredit Group Performance 2007

The following are average values of availability in 2007 for the main services UniCredit Group provided to Customers.

Table 3. UniCredit Group service availability performance (2007)

Service Type	Window	Range of Avg % Availability (1)	Note
Large Workload Banks	11 x 5	99,21 - 99,70	Services to Branch Offices. Large Retail banks of the Group.
Small Workload Banks	11 x 5	99,44 – 99,86	Services to Branch Offices. Smaller specialized banks.
Internet services	20 x 7	99,21 – 99,69	Comparable to Branch Offices Services
ATM/POS	24 x 7	99,57 – 99,80	Included planned maintenance.

(1) Included planned outages

The work done over the year in terms of awareness of persons and tools, and the unrelenting improvement in the day by day activities, led us to improve dramatically the level of quality of the service provided to Customers. The following graph represents our history related to Service availability level, and witnesses how our efforts have been rewarded by high level results.

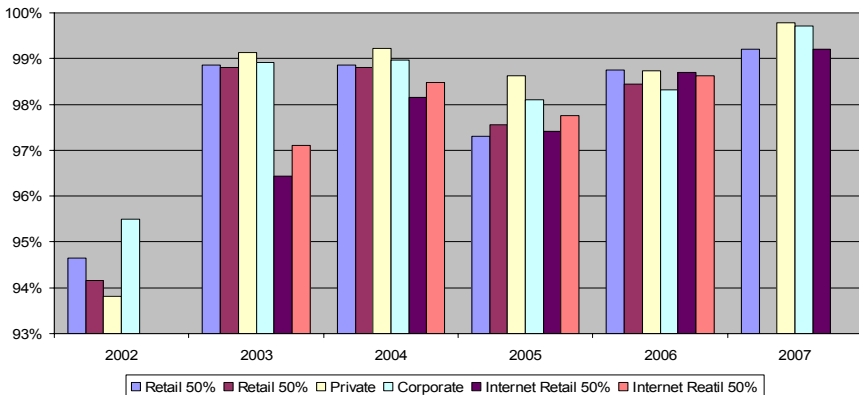


Fig. 2. UniCredit Group service availability trend since 2002

UniCredit Group Positioning

Gartner [6] provide some forecasting and ranking related to availability. We use it as a benchmark for us.

Table 4. UniCredit Group positioning for service availability

Gartner Ranking 2007	Only Unplanned		Planned included		Window	
	h/year	%	h/year	%	h/d	d/w
Best in Class	5	99,94%	12	99,81%	24	7
Outstanding	26	99,70%	50	99,13%	24	7
Very Good	61	99,30%	200	97,02%	24	7
Average (less than)	175	98,00%	250	95,15%	24	7
UniCredit Group						
Retail	UniCredit Group includes Planned downtime.		23	99,21%	11	5
Private			6	99,78%	11	5
Corporate			9	99,70%	11	5
Internet Retail			58	99,21%	20	7

The positioning of Unicredit Group is higher than “Very Good”, and in selected services better than “Outstanding”.

5. Considerations and Conclusions

The situation described must be a stimulation to adopt and engineer a “continuous improvement” approach.

On one side, to press and to work closely with the supplier to demand a higher products quality. On the other side, to put in place an unrelenting effort to improve application delivery, change management, incident management and problem management processes. The hard reality, in fact, is that in the next coming years a given (possibly high) amount of faults will hit IT Service Providers with growing trends (many devices/products will reach the obsolescence before the maturity and stability). Given that, the ICT management attention must focus on following issues:

1. The structuring and organization (re-engineering) of processes related to: Change Management, Incident Management, Problem Management, Customer/User Communication are the most important critical success factors for an ICT Service Provider in the next coming years. Even though functions are perfects, if the Service is not outstanding the perception of the Customers will be very bad.
2. The technology adoption and upgrade must be carefully evaluated, balancing innovation with technology risks; a “follower” policy in the adoption of new technology can lower risks vs a delay in new products, face to face with an “early adopter” policy that can raise risks vs an earlier market innovation or capacity to process growing traffic volumes. The Customers, i.e. the business should be informed and aware of this balance.
3. The capacity of the organization to face the faults in a proper and timely manner comes not only from structured processes, but mainly from a

“sensitivity/attitude” focused to “recovery the service” (market oriented approach) and not to “solve the problem” (technical approach). Incident should be overcome even with a work-around; definitive solution should come after, opening and managing a problem (asynchronous activities in respect to incident management). In addition, some organizational and procedural actions can help reducing incident duration; in our experience the main areas of improvement are:

- to enable early detection of incidents via probes and monitoring tools;
 - to simplify the structure of the service chain (less instances of Operating Systems, Virtualization, ...);
 - to provide more effective and possibly automatic procedure to 1st level support;
 - to provide better procedure for failover of redundant devices;
 - to provide more effective restart procedures.
1. The availability percentage, in itself, is meaningless, if not complemented by the approach, the method and the algorithm which it is calculated with. E.g. the old approach of measuring availability of sub-systems and components (mainframe, servers, network...) leads to completely different figures.
 2. The availability percentage is aimed, besides being a communication tool to the customers, most of all as a tool to understand what is happening, and to direct the attention of personnel to the most frequent causes of faults. Trends are much more important than the figures itself!
 3. At the time being, there are theoretical limits to the target that can be reached; it is important to define them in each own business and operating environment, with the aim to compare the actual performance with reference benchmark. The measure must be focused:
 - to improve the ability to react, both for causes “embedded in technology” and in those due to human errors;
 - to log and document actions taken, in order to identify possible opportunities for shrinking process times;
 - to give elements for technology and organization evolution;
 - to acquire and capitalize experience (knowledge management).

References

1. Martin J (1978) *Systems Analysis for Data Transmission*. Prentice Hall PTR, New York.
2. Sheehan J (2006) Understanding Service Sector Innovation. *Communications of the ACM* 49, Issue 7:42-49
3. Van Bon J (2007) *Foundations of IT Service Management Based on ITIL® V3*. Van Haren Publishing
4. Isermann R, Ballé P (1998) Trends in the application of model-based fault detection and diagnosis of technical processes. doi:10.1016/S0967-0661(97)00053-1
5. D'Ambros M, Lanza M (2006) *Software Bugs and Evolution: A Visual Approach to Uncover Their Relationship*. doi: 10.1109/CSMR.2006.51
6. Scott D (2007) *Operation Zero Downtime*. Gartner Group report