To The Editor:

In the article "Proposed Guidelines for an Automated Cartridge Repository" (Computer, July 1985, pp. 49-58) author Patric Savage states: "For example, if pickers have a mean time to failure (MTTF) equal to 365 days and a mean time to repair (MTTR) equal to one hour, then for two pickers the composite MTTF = 365 days = 365 years (approximately)..." I feel this could be misleading to the readers of Computer.

If Mr. Savage is assuming a parallel redundant configuration with two elements (or components), which is the situation referred to in the graph in Figure 2, the MTTF of the parallel configuration, MTTF_p, can be found by integrating the reliability function of the configuration

\[ MTTF_p = \int R_p(t) \, dt \]

The reliability function for a parallel configuration with two components is

\[ R_p(t) = 1 - [1 - R_1(t)][1 - R_2(t)] \]

with \( R_1(t) \) and \( R_2(t) \) the reliability functions of the components. If the reliability of the two pickers is equal, this is

\[ R_p(t) = 1 - [1 - R(t)]^2 \]

where

\[ R_2(t) = R(t) \]

The line for two elements in the graph of Figure 2 is the result of this equation, with the reliability of the components assumed to be constant over time. This is equivalent to assuming a constant failure rate with respect to time, and leads to the use of an exponential distribution to model the component failure probabilities. For this distribution \( R(t) = e^{-\lambda t} \) where \( \lambda \) is the constant failure rate and \( t \) is the time variable.

Substituting this into Equation 3 above gives

\[ R_p(t) = 1 - [1 - e^{-\lambda t}]^2 \]

Putting this into Equation 1 gives

\[ MTTF_p = \int [1 - (1 - e^{-\lambda t})^2] \, dt \]

After a little integration this gives

\[ MTTF_p = \frac{3}{2\lambda} \]

Since the component

\[ MTTF_c = \frac{1}{\lambda} \]

for this distribution,

\[ MTTF_p = \frac{3}{2} MTTF_c \]

is the relationship between component MTTF, MTTF_c, and the MTTF of the parallel configuration, MTTF_p. For the specific example of MTTF_c = 365 days, MTTF_p = 1 1/2 years.

If, on the other hand, Mr. Savage is assuming that the pickers are repairable, as the mention of mean time to repair in the text indicates, there is a different result for the case of the exponential distribution. Then

\[ \mu = \frac{1}{2\lambda}, \gamma = \text{the mean time to repair} \]

where \( \lambda \) is the failure rate of the two components and \( \gamma \) is the mean time to repair. This can be represented as

\[ \mu = \frac{1}{2\lambda} \times (8760)^2 \approx 3.84 \times 10^3 \text{ hours, or 4380 years.} \]

I am not sure which analysis Mr. Savage was thinking of when he wrote the example. The two situations lead to widely different results. Looking at the site configuration, I would have chosen the second situation with the repairable pickers as an example.

Shaw Matthews
Storage Technology Corporation
Louisville, Colorado


Author's response:

We wanted to leave the impression that redundancy gives such extreme composite reliability that sites can rule out failure as a consideration, but felt that erudite mathematical derivations of reliability would be quite out of place in our paper. That is why we made up an example that we hoped would be intuitively easy to understand. We also deliberately used an old (1963) reference to suggest that this is well-established knowledge. Of course, we were assuming repair—it should be obvious that we were applying the product-of-the-probabilities formula as a crude approximation. We used 365 days, not for realism, but because it is one year. It is the MTTR = 1 hour that made our "365 years" wrong. Here is a restatement of our example:

For example, if pickers have MTTF of 31 days, and MTTR less than 2 hours, then for two pickers the composite MTTF equals (31 x 12)^2 two-hours = 365 years (very approximately).

Patrick Savage
Shell Development Company
Houston, Texas

To The Editor:

In the interesting issue on visual programming, (August 1985) Georg Raeder discusses several graphical pencil-and-paper software design methods and assesses their potential for animation.

One that is not mentioned is Jackson Structured Programming, a highly graphical design method based on data structures. It is reasonably well-established for business applications but applies to programming in general. I have been using it successfully in systems programming for many years.

In JSP, program logic is expressed in structure diagrams, which would lend themselves elegantly to animation. During program execution, the boxes in a diagram would be highlighted successively as control moves through the structure. This would be particularly interesting since the current location of control in a JSP program carries a maximum of information of the current state of the program, while the use of state variables is kept to a minimum.

JSP design often results in cooperating programs, such as parallel processes or coroutines, working on the same data. In this case, two or more structures could be displayed simultaneously, showing how control moves through each independently.

Some graphical JSP design tools exist but, to my knowledge, animation of program execution has not been attempted. The reader may get a first acquaintance with JSP in my article in the October 1985 issue of CACM. Further references are listed there.

Bo Sanden
BANPAIS
Monterrey, Mexico