

Re: Productivity

Source: <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2007-01/msg00000.html>

- *From:* "Pete Dashwood" <dashwood@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 1 Jan 2007 14:23:22 +1300
-

"LX-i" <lx0007@xxxxxxxxxxxxx> wrote in message
[news:ce747\\$4597ebb6\\$454920f8\\$18039@xxxxxxxxxxxxx](news:ce747$4597ebb6$454920f8$18039@xxxxxxxxxxxxx)

Robert Jones wrote:

message snipped

Pete Dashwood wrote

This, admittedly contrived, example demonstrates quite clearly the fundamental difference between event driven (read, "on-line") and non-event-driven (read, "batch") processing. OO lends itself to event driven processing; procedural programming doesn't. That is why OO is taking over the world... (whether the people using it understand this or not... all they know is that Objects are flexible and responsive and designed to handle events; procedural code simply isn't...)

While I can believe that COBOL will eventually be retired, I don't believe batch processing will ever completely go away. Events such as are invoked by as tax authorities wanting yearly company accounts and tax computations are probably too much to suddenly throw at an on-line system, there would probably be severe degradation of service, even if many of the underlying figures had already been calculated. Events such as monthly/weekly payrolls, cheque printing, etc also are probably best done in a batch process.

Re: Productivity

This is a good question. How would these processes be handled in an "on-line" way? I know you wouldn't have to actually have to calculate payroll until it was time to generate the direct deposit/print the che[ck|que], but doing this serially for all active employees on a certain date would seem to be, at best, "online with a driver."

Pete, What would be your online vision of how this would take place? (You know, once you're done with the beach party and all...) ;)

Sorry, I responded to Robert before I read your post, Daniel.

I agree that payroll is a very good example of a process that is handled well by batch processing, but I also don't think it HAS to be.

I remember many years ago working on a payroll system for Auckland Hospital. The design took nearly two years (I didn't do it – I was a programmer). It had to cater for every conceivable occupation from janitors to senior surgeons, all with their own special awards, penalty rates, holidays, payment methods, and payment periods. Just getting information into it (who was on leave, who had been promoted, who needed a final pay because they were leaving, how many hours were worked by people who were paid hourly, etc. ad nauseum) kept the Payroll department fully employed just entering transactions to notify it of exceptions to the underlying base.

If you want a vision here is some holiday induced fantasy... ;-)

Nowadays, I believe all of the base processing could be achieved by stored procedures. These procedures could be triggered against each employee account on a staggered schedule (say, one every minute, or whatever, depending on the number of employees. – the hospital had around 700, so, if you allowed 1 day at the start of the month (say the first banking day of the month) you would do a base calculation for an employee once every two minutes. (Hardly a noticeable load on the transaction system.) The result of this would be a schedule of payments to go to the Bank on Payday, with a line for each employee, keyed on their employee id. During the month, transactions for exceptions would occur and each of these would modify the line on the schedule for the affected employee, as well as updating their account, tax, deductions etc. Come Payday, the schedule is already prepared and reflects up to the minute transactions and events. It is sent to the Bank, everyone gets their money... Hooray!

Obviously, this is simplistic and assumes that everyone is paid on the same day and in the same periodicity, and by direct credit, but it could be modified for exceptions to this. For example the schedule could have an indicator for the payment method, and trigger a cheque instead of a transfer. Separate schedules could be prepared for people who are paid weekly, monthly, hourly, whatever. It is feasible and it doesn't require daily batch runs or even monthly batch runs with the consequent impact on

Re: Productivity

batch windows.

Will it happen? Extremely unlikely, but it could... A small business with a small number (say, 50) of employees might find this a very useful way to "automate payroll and put it on a back burner".. Because it is on-line it would reflect commission payments into the period they were earned (very popular with the salesmen), rather than waiting until next month because the batch cut-off was missed.

Having said all of the above, I have to agree that payroll is one example of something that is best addressed (at least right now), by a batch process.

Pete.

.