

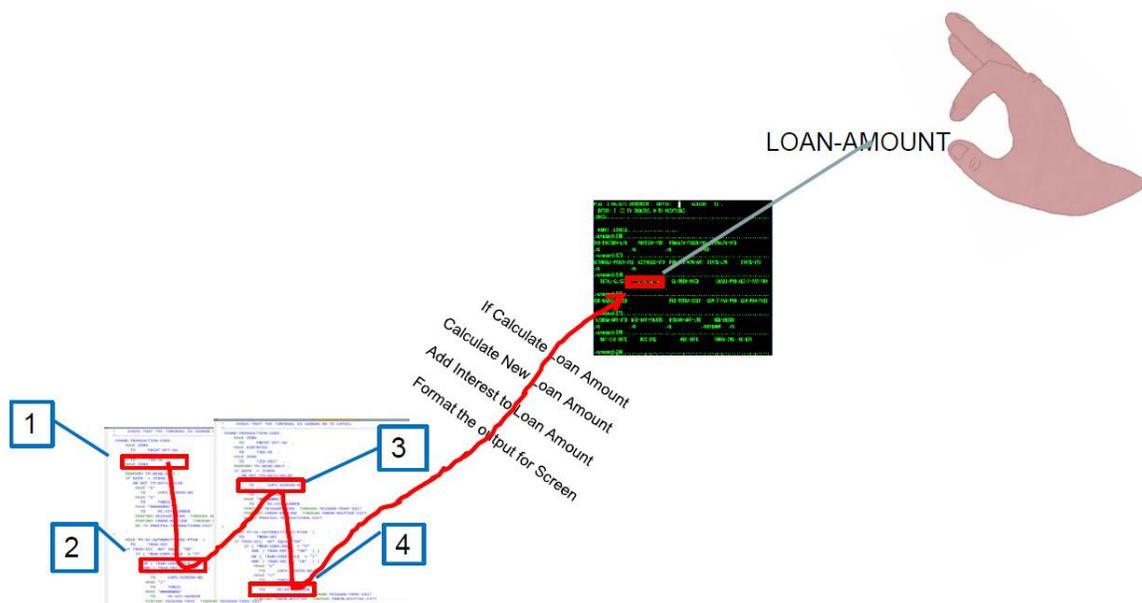
WHY IS IT SO TOUGH TO DETECT BUSINESS RULES?

Denise P. Kalm

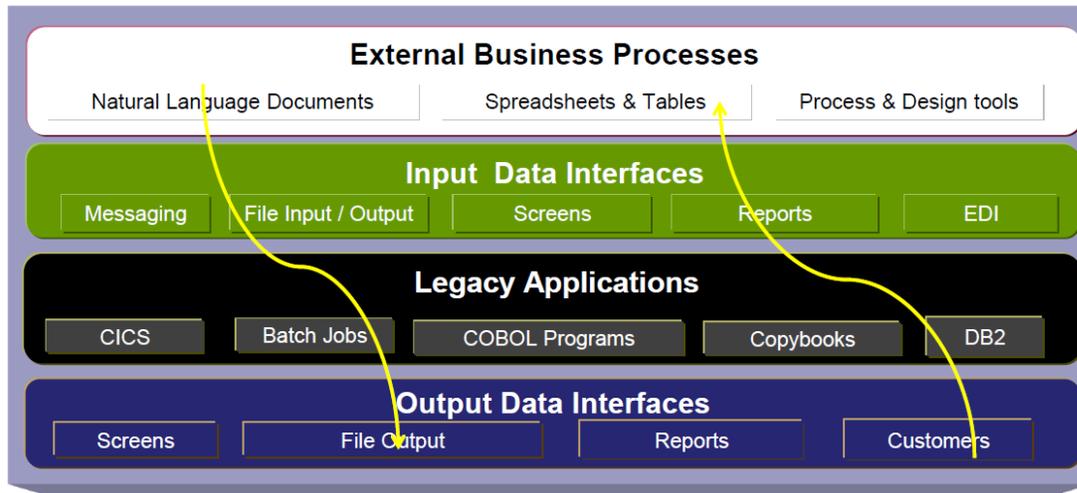
Want to modernize your legacy applications? The first step is to understand the business rules underlying the code, so you can ensure that you don't lose any functionality. Once, this might have been easy, when applications were simple and one program handled all the logic. But that's not the state of affairs in most enterprises. Applications are complex beasts spanning programs and platforms and even geographies. Documentation, if it exists, is likely not to be up to date. And most programming languages make it difficult to parse out the code that represents a business rule from the code that makes a program work.

Say the business rule is the set of conditions under which you might send a letter to a customer. "Send a letter if the customer is two weeks late on payment of a bill." It sounds like a single line of code might handle this, but there's more to it than that. How do you detect that the payment is late? You might run a batch job regularly to pull out customers who haven't paid. But then, you need to create the letter, pull an address, maybe tailor it based on prior bill paying performance, etc. You also need to record that you've sent the letter and set up actions if they fail to pay in the next window. In many cases, the code began simply and over time, grew more complex. In fact, you might have situations where billing service operators can open a dialog and request a letter be sent. So you have presentation logic on their PC to handle that. How do you follow that thread to be sure you've hit every logic point? But if you want to modernize the code, you have to know where every piece of it resides. Anything you miss means the resultant code won't produce the same results as the legacy application.

Now, consider that this example is only a small part of all the logic on your system. If you are becoming daunted at the impossibility of the task of documenting the logic, you're not alone. But what if you could actually 'pull on a thread' and see the connections for each piece of logic across programs and platforms? And what if you could visualize it in a variety of ways, so you knew that when you touched a piece of code, you understood all the implications of changing or modernizing it?



It turns out that it is completely possible to have this magic thread and create complete documentation, automatically. CM evolveIT can draw the thread through allied processes, defining the connection between your business rules and the underlying code.



It works through a patented process to ensure complete and accurate capture of your rules, creating flexible documentation options you can use to understand and modify your code. The process works as follows:

1. Model the application using CM evolveIT automated capture
2. Identity application-related business processes
3. Identify data in and out of the application for each business process
4. Define the business terminology for input and output data
5. Complete the data logic trace within the application
6. Combine to build business rules.

CM evolveIT is the missing piece of your modernization puzzle. Contact CM First to see a demo. And check out the recorded webcasts mentioned elsewhere in this newsletter. See how this solution can make the impossible possible.