



Processes in Practice

Rob Davis

Rob.Davis@ids-scheer.com

If you are not Modeling Data how do you know your Processes Work?

When people first started to seriously think about designing, modeling and analyzing processes they were mostly concerned with manufacturing processes. The processes were a description of the work to be done and this was largely about the physical work of making an item, installing it, and fixing it. Of course, information was needed to manage these processes, but most of the focus was around the physical tasks.

Today, processes are much less focused on physical tasks. The reasons for this are two-fold. Firstly the rise of services industries has meant that processes, although still requiring manual tasks, are more focussed on delivery of contracts and services and less focussed on actually manufacturing a physical item. Secondly, the impact of IT on all businesses has tended to result in more abstract or conceptual processes where people, instead of doing physical tasks, manipulate data. This data manipulation can either be the end-in-itself for a process where nothing is physically produced (e.g. processing a mortgage application) or it is the driving force for managing the physical tasks that may seem far removed from the process operators (e.g. scheduling visits to customers).

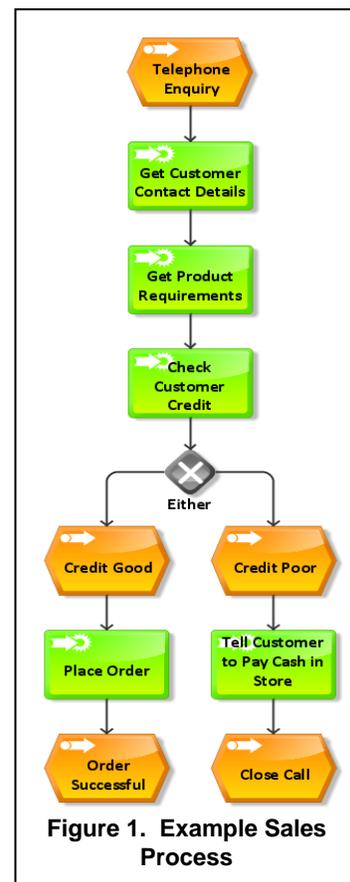
The end result of these two factors – service-driven industries and IT – is that rather than describing how physical resources are transformed and moved through the process, most processes now describe how people and systems manipulate data. Given this situation, the surprising thing is that very few of the process models that I come across actually mention data at all! Furthermore, rarely do people test their processes by “walking” test data through them. So my question is:

“If you are not modeling your data how do you know your processes work?”

Why Model Data?

Consider the simple sales process in Figure 1. A customer rings our business to order a product. We take their contact details and find out what product they want. Let's assume we sell high value products so we need to check that their credit is good. If it's OK we place the order; if not we tell them they need to go into one of our stores and pay cash.

This process looks straightforward and is typical of the way that many people model their processes. The question is will it work? Walking through the process steps it looks fine, but now let us



look at the same process again, this time with some of the data modelled (see Figure 2.).

The blue symbols represent formal data that is available to the process (I have simplified it and assumed we know data such as the customer’s name).

In the first step we capture the customer’s address and telephone number and then we identify what product they want. We then carry out the credit check for which we need the total cost of the goods they want to buy and their credit card details. If all is OK then we place the order for which we need their address and the product reference.

Again this is typical of how people model and at first glance it looks fine, especially if we imagine that we are doing all these steps manually and we can ask the customer for any additional information we need while they are on the phone,

However, let us assume that we can only capture customer information in the first two steps and the credit check is done automatically based on information we have captured. Even a quick glance at the process may suggest that something is wrong and the process will not work.

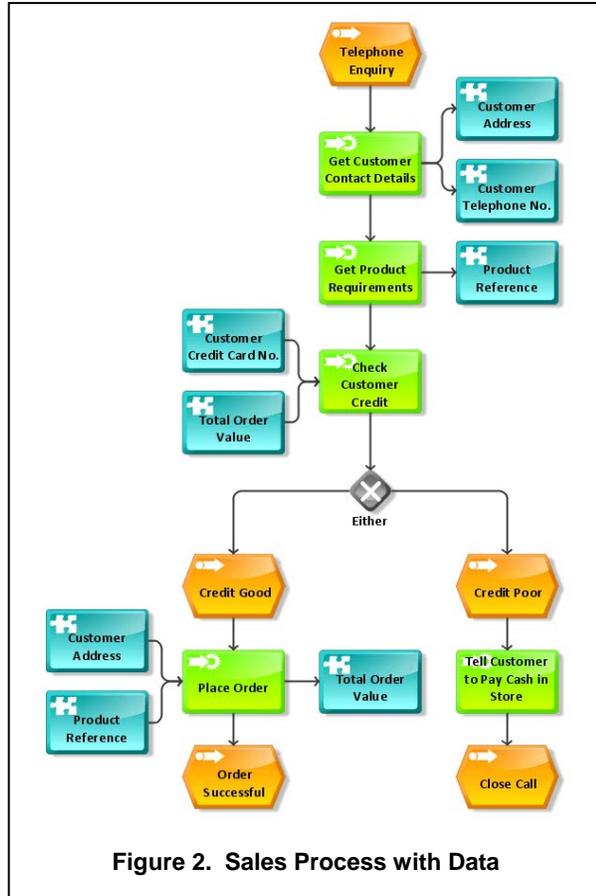


Figure 2. Sales Process with Data

Process Testing Using Data

To more formally test if our process will work, we can undertake data analysis to see how the data is being used. A popular and very useful technique is CRUD analysis. “CRUD” is a way of representing the Create, Read, Upsert and Delete actions carried out by the process steps on each item of data. Figure 3 shows this analysis for the process in Figure 2; again simplified as we only have create and read actions.

<i>Data</i>	<i>Customer Address</i>	<i>Customer Telephone No.</i>	<i>Product Reference</i>	<i>Total Order Value</i>	<i>Customer Credit Card No.</i>
Get Customer Contact Details	C	C			
Get Product Requirements			C		
Check Customer Credit				R	R
Place Order	R		R	C	

Figure 3. CRUD Data Analysis

On the left-hand side, the row headers list the steps or tasks in the process. At the top, the column headers represent the data entities. The corresponding cells show the appropriate action carried out by the process steps on the data. In this simple example there is only one action for each step, but it is possible to have more than one action. For instance, a step may Read an item of data and then Uppdate it so in this case you would see “RC” in the appropriate cell. If you are using a process modeling tool to design your processes, then most tools can produce a CRUD analysis automatically. If you are drawing your process, for instance using Microsoft Visio, then you will have to create the CRUD manually.

The first check we can carry out is to ensure that every piece of data that is to be Read by a task has been Created by another task. Some people can get confused by what is meant by Read in a manual process. It is tempting to see Read as the action of asking the customer for information, but in the formal data modeling sense Read means to access a piece of data that is stored in an IT system (or is at least already known to the process – it may be written down). So a task that asks the customer for information Creates data (in an IT system or on a piece of paper) and later tasks Read that information.

If we look at Figure 3 we can see that “*Customer Address*”, “*Customer Telephone No.*” and “*Total Order Value*” all have Read and Create status, but that “*Customer Credit Card No.*” has only Read. So we need to use credit card information in the process, but we haven’t asked the customer for that data. You might think this is not important; why not just ask them for it? Well, of course, in a simple manual process you can, but in more complex processes where a lot of information is needed it is important to capture the right information at the right time. The customer will get annoyed if you keep going back and asking them for more and more pieces of information in a random manner. Furthermore, if the process is to be automated or implemented as part of an e-process on a web page, then you can’t go back and ask for pieces of information you have forgotten; you have capture all the information at the correct time.

We can easily amend our process to ask the customer for their credit card details. Of course, we can’t just add that to the initial “*Get Customer Contact Details*” step. The customer is going to be a bit suspicious if we ask them for their credit card details right at the start of the dialogue before they have even chosen a product. More likely, we will add an additional step “*Get Customer Credit Card Details*” after the “*Get Product Requirements*” where we explain that, because it is a high value product, we need their credit card details to do a credit check. If the customer is seriously interested in buying the product at this point they should be happy to provide these details.

We can also look for data items that only have Create actions. This means that the data is being captured, but never used. This doesn’t cause the process to fail, but it may represent inefficiency in the process and may also require IT to store unnecessary pieces of data. It is quite common for redundant data to be captured in processes; maybe it was used at one time but isn’t used now and nobody has removed the need to capture it.

By itself, the CRUD analysis does not tell us everything we need to know to test the process. It is not sufficient to just check that every data item has a Create and Read action, we must also look at the sequence of those actions. It is obvious that the Create action should come before the Read action. In the simple CRUD in Figure 3 I have listed the steps in the order that they occur. You can see from this that the “*Total Order Value*” data is actually used before it has been created. To do the credit check, the value of the order is needed, but this is not available until the order has been placed. This is a serious problem and we will need to refine the process again.

As shown, the “*Place the Order*” step enters the required order information including shipping details and from that it can calculate the “*Total Order Value*” data. Instead what we need to do is split this task in two. Firstly an “*Enter Order*” step after “*Get Product Requirements*” enters all the product information and calculates a total order value excluding shipping. The “*Total Order Value*” data is now available for the “*Check Customer’s Credit*” step. It also makes sense to enter all the product information at this point in the process as we have just elicited it from the customer. The “*Place the Order*” step now only enters shipping details, then confirms the order and updates the “*Total Order Value*”. The revised process is shown in Figure 4.

The Value of Data Modeling

We can now see that by considering the dataflow we found some problems with our initial process and have arrived at a much better result. The example I have given is simple and somewhat contrived, but I have in fact used dataflow analysis for real on a web-based sales order process and detected similar sorts of problems. In particular, the issue of when to do credit checks and when to collect customer financial information became very complicated. Once we did the dataflow analysis we found there was a significant difference between the way the marketing and sales people wanted the process to work and what was practically possible given the data requirements. These problems had not emerged at the initial process walkthrough stage when data hadn't been considered.

Corporate Data Models

Another issue that becomes apparent when you consider data in processes is how little people understand and agree about what data a business uses. One might imagine that every large business has a corporate data model that identifies all the key business data and defines exactly what it means. In my experience few businesses have this, even when the business has a large and mature IT department.

When I undertook the dataflow analysis on the web-based sales order process mentioned above it became clear that nobody really understood the definition of a “customer”. The sales and marketing people informally defined a customer as anyone who was interested in buying a product or who had already bought one. For the IT department, someone was only a customer after they had bought something. This distinction became important when designing the web-based process because a person browsing the web site might be completely anonymous, or they might have previously registered on the site and logged in, or they might be a returning customer.

Depending on which category of person was following the web-based process determined what information we already knew about them. It also determined when we could ask them for specific pieces of additional information and when we could do a credit check. Furthermore, a person might “change state” during the process if they later logged in. Most of us are familiar with web-sites such as Amazon.com where you can login at the start of a session or you can order something and then login.

State Diagrams

The experience of the customer data definition issue made us realise that as well as having CRUD states, the business entities (e.g. customer) could change state as the process progressed or over longer time periods. Instead of a simple definition of customer, we then developed the more general concept of a “Party” (someone interacting with the business) who could change state: from “Browser” to “Contact” to “Customer” to “Major Customer” (I have simplified this again). The sales order process itself also changes state as the “Party” progresses through selecting items, providing information and then finally committing the order.

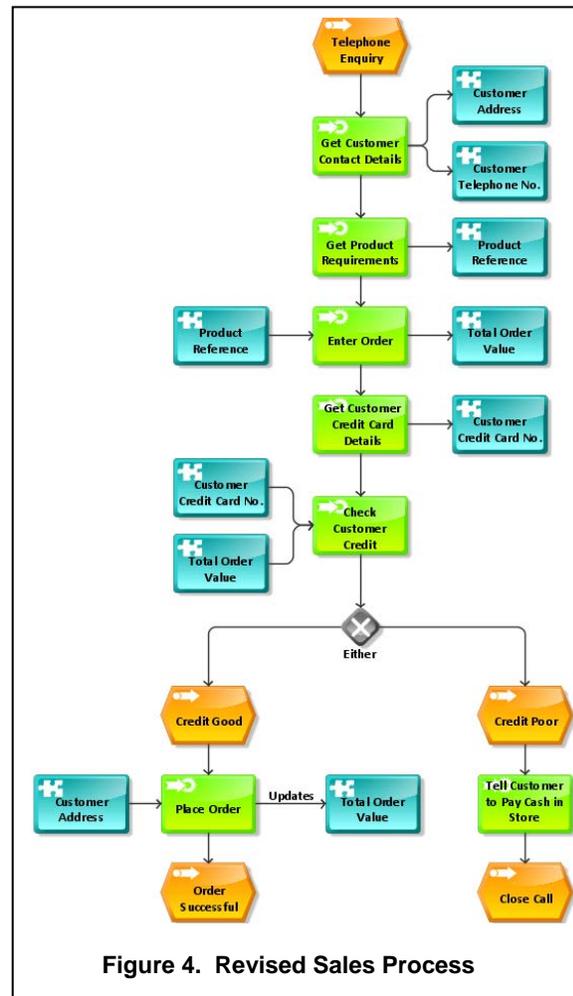


Figure 4. Revised Sales Process

State Diagrams or State Transition Diagrams are well known techniques in engineering and software design which can be valuable in process design, especially when the processes are to be automated or web-enabled.

Data Scenario Walkthroughs

In real processes, it is not possible to do the simple CRUD analysis I showed in Figure 3 and list the steps in sequence. Complex processes have many paths and branches so working out whether Create comes before Read is more difficult as it depends on the specific path traversed through the process. It is therefore necessary to define various test scenarios that walk different routes through the process. These scenarios can then be exercised by either manually walking the process or by using business simulators or test scenario tools. With a very complex process it may not be possible to test every possible combination, so it is important to choose a representative set of test scenarios that check the most likely and important paths.

Processes Need Data

In this Column I have only been able to touch on data modeling techniques, but hope I have been able to illustrate how modeling data and data flows is not just beneficial to process design, but is absolutely essential to ensuring correctly operating processes.

I believe that the need to include these techniques as part of process design highlights the fact that process design and modeling is a professional occupation. It should not be left to the summer student or Intern using a drawing tool, but requires professional people and professional tools and techniques. After all:

“Process is not just something your business does – process is your business”

I hope you will consider modeling data in your next process project.

Author

Rob is a Principal ARIS BPM consultant with Software AG which merged with IDS Scheer in 2011. He is an internationally recognized expert in Business Process Management (BPM) and the practical use of the ARIS Design Platform. Previously, Rob worked for British Telecom (BT) where he was responsible for selecting and implementing ARIS in a large scale implementation. Rob has built extensive experience of all aspects of BPM and specializes in providing consultancy on BPM, process modeling and design, architecture and frameworks, process governance, and integrating process and IT design. Rob has written three definitive books on the practical use of ARIS Design Platform for BPM.

For more information see <http://www.rob-davis.co.uk>.

BPTrends LinkedIn Discussion Group

We created a BPTrends Discussion Group on LinkedIn to allow our members, readers and friends to freely exchange ideas on a wide variety of BPM related topics. We encourage you to initiate a new discussion on this publication, or on other BPM related topics of interest to you, or to contribute to existing discussions. Go to LinkedIn and join the [BPTrends Discussion Group](#).