

# White Paper from Global Process Innovation

by Jim Boots

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

This white paper continues the debate begun in *Chapter 4: BPM Standards and Methods* in my book *BPM Boots on the Ground* that favors Universal Process Notation (UPN) over Business Process Modeling Notation (BPMN) in the descriptive layer of process representation. Let's look at an example of a BPMN process model reproduced from Bruce Silver's book and compare that to the kind of process model we might create in UPN.

Figure 1 contains a reasonably accurate facsimile of Mr. Silver's *New Car Sales* process as shown in his book, "BPMN Method & Style".

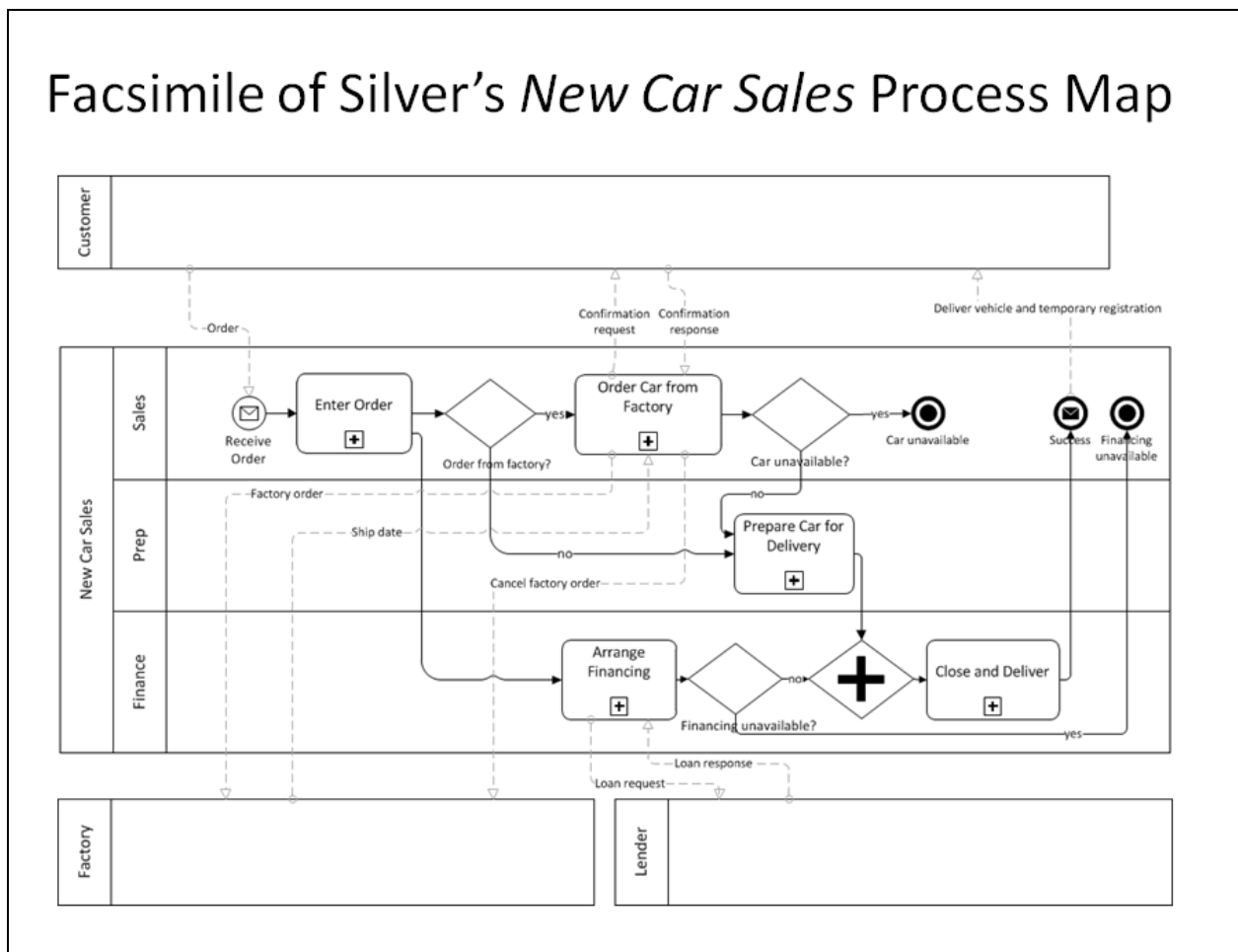


Figure 1: *New Car Sales* process map from Bruce Silver's book, "BPMN Method & Style," 2009

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

For a side by side comparison, Figure 2 contains a rendering of the same process in UPN:

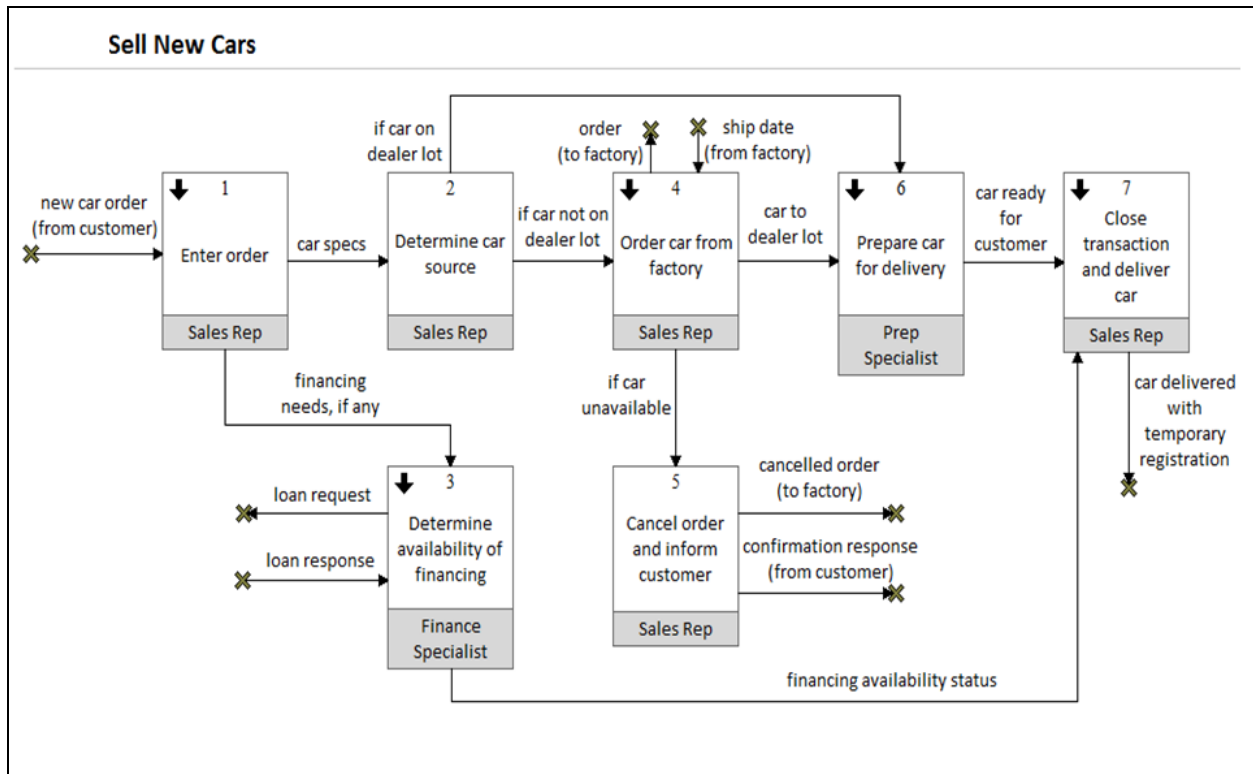


Figure 2: Translation of Bruce Silver's *New Car Sales* process map into Universal Process Notation

As we compare the representations of this New Car Sales process in the two notations, let's start with a few statistics in Table 1 to see if they shed any light (with apologies for any minor miscounts):

Table 1: Comparative statistics between diagram types that represent a similar process

Visual Dimension	BPMN	UPN
Number of activity boxes	5	7
Number of decision diamonds or gateways	4	0
Number of symbol types	11	5
Number of words	63	101
Number of swimlanes	6	0
Number of flow lines	23	16
Number of crossing flow lines	6	0
Number of non-straight flow lines	16	3

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

Through examination of these statistics alone, we may conclude that BPMN is structurally more complex than UPN, and that to convey meaning UPN seems to rely more heavily on words rather than symbols. These attributes will tend to be true on almost all equal comparisons between BPMN and UPN. UPN relies more on words (which humans can easily interpret) rather than symbols (which machines can easily interpret). But UPN does not totally reject symbols, either! UPN takes advantage of a few symbols that are most intuitive and that add value in showing key relationships between the words. Some will argue that if a *few* symbols are good, then *more* symbols must be better. The difficulty with this is that the more symbols there are, the less intuitive they are and the more complex the potential relationships between the symbols. Plus, you also have to remember the meaning of each symbol. This is never an issue with UPN. UPN is designed with humans in mind so it aims to optimize the value of symbols. Like Goldilocks (in the Story of the Three Bears), UPN is looking for the porridge that is “just right.”

Moving to more of a qualitative comparison, we note that UPN attaches resource names to the activity boxes rather than using *swimlanes*. Swimlanes, along with *decision diamonds* (which I address next), are particular favorites of BPMN practitioners. However, in UPN we simply label each activity with the resource (or resources) that are needed for that activity. For a straight-through process where there is only one resource involved, a swimlane will, in fact, be slightly more graphically economical than a UPN diagram with the resource shown on each activity. But, how many processes are going to be that simple? Even in our fairly simple New Car Sales example, we can see that the UPN diagram looks much cleaner—with more white space—than the BPMN version. Furthermore, if we decided that a Sales Manager instead of a Sales Representative was to order the car from the factory, and we wanted to distinguish these roles, we would have to add another swimlane to the BPMN diagram. This would crowd the BPMN diagram considerably whereas in the case of a UPN diagram we would simply change the resource name attached to box 4.

There are other, subtler points as well. When rendering Silver’s BPMN diagram into a UPN diagram, I was somewhat inconsistent with the naming of resources. Silver kept the resource names generic (i.e., Sales, Prep, Finance) whereas I assigned actual positions within those organizations (i.e., Sales Rep, Prep Specialist, Finance Specialist). The difference in the example shown is fairly insignificant (Silver could have used my more specific resource designations, or I could have used his more general ones) but this difference is indicative of a difference that becomes important in more complex processes. In a more complex process involving lots of different individual resources, the number of swimlanes can become challenging—leading to diagrams with lots of swimlanes jammed together, or multiple pages, either of which makes discussion of the processes much more difficult.

Another complicating issue arises with swimlanes when multiple resources are involved in a given activity. An organization can make rules to prevent this if it wants, deciding, for example, that only one resource should be assigned to any given activity. But my experience is that people are often involved in collaborative activities that involve more than one resource. Representing this on a BPMN diagram adds more structural complexity, but on a UPN diagram it is simply a matter of appending another resource box to the individual activity. To summarize: in general, BPMN diagrams take up a lot more space than UPN diagrams.

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

Swimlanes also require “*matrix thinking*.” This is not difficult for most analysts or engineers, but it is not a natural way of thinking for everyone. Abstracting the “*who*” from the “*what*” may be useful for analysis, but it is not necessarily helpful for people when they are first thinking about a process. For example, in a process documentation session using UPN, for each step in the process the modeler can ask the SMEs, “What happens? Who does it? What are the outputs?” The answers to these questions will all be captured in the Universal Process Unit (UPU) construct. However, in a BPMN-based session the modeler has to keep adding swimlanes when new resources arise as the process unfolds, and invariably the swimlanes will need to be re-orientated so arrows do not intersect haphazardly with swimlanes throughout the diagram.

Another subtle point: swimlanes confuse *the visual flow* of a process. Due to the necessity to insert boxes into swimlanes when using BPMN, any complex process that has three or more interacting activities will have numerous twists and turns. By visual examination, look at the simplicity of the arrow flow lines in the UPN diagram compared to the BPMN one. The last statistic shown in Table 1 (i.e., number of non-straight flow lines) is a bit inflated for the BPMN diagram beyond what it needs to be because some of the flows could be made straight if desired. Nevertheless, if we focus attention on just the arrows in the BPMN diagram, there certainly seems to be considerably more *flow complexity* versus the UPN one.

A final subtle point is the increased sense of accountability that is conveyed by attaching the resource directly to the activity. This probably seems insignificant as there is no theoretical difference. Nonetheless, when a person’s role is attached directly to the activity, there is a greater sense of ownership versus having the activity located within a swimlane that is assigned to the role. With UPN, the activity itself can be abstracted from the diagram—for example, for training or for job aids—but the performer still will see her role attached to the activity, whereas the swimlane assignment might be lost. Also, when capturing or innovating a process, there is less hesitancy to simply change a resource name attached to an activity box rather than move the activity box to another swimlane with the attendant re-routing of flow lines and message arrows.

So, in summary, for descriptive modeling, UPN’s use of resource boxes attached to activity boxes is superior to BPMN swimlanes for the following reasons:

- Less visual complexity and more room for important content
- Easier live capture of process models
- More visually elegant process flows
- Enhanced sense of accountability for activities

Now, let’s address the decision diamond—the other sacred cow of BPMN practitioners. Let’s start with what I consider the only legitimate advantage of decision diamonds—the unequivocal visual message that “*this symbol represents a decision*.” How much of an advantage is it? In the human world, is the ability to distinguish decisions from other activities important?

In the machine world, decisions have a special meaning because they are necessary for communicating pathways within binary systems; therefore they are called out as *gateways* where the subsequent path

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

is determined based on given criteria. Other non-decision activities do not require a choice based on criteria; rather, they simply do everything specified, and all paths out of the activity are followed. So, in the machine world, the different shape, a diamond, specifies a different kind of activity.

But, in the human world of non-automated process steps, is there really so much distinction between a decision and any other activity such that a decision needs to have a different shape? In fact, to a human, isn't a decision simply an action—one that can be described by starting the activity statement with the word "Decide" or "Determine"? For example, in our UPN-based *Sell New Cars* process in Figure 2, is the activity "Determine car source" any more important than "Order car from factory" (assuming the desired car was not on the lot)? In fact, we probably could make the case that the "Order car from factory" activity is more important than checking whether the desired car is on the dealer lot because, if the decision step is skipped and the Sales Rep jumps right to ordering the car from the factory, at least the customer will get her car. Whereas, if the car is not on the dealer lot, and if the car is never ordered from the factory, then the process will not complete and the customer will not be satisfied.

To take another example from oil exploration activities, the choice of a contractor to collect seismic readings from a field might be considered an important *decision*, but it certainly is no more important than the *activity* by the oil company's geologists of interpreting and evaluating those seismic readings? In other words, to humans, decisions are no more important than other activities.

Some practitioners argue that decisions have a special meaning as control points; that is, points around which there should be tracking and governance. But, for example, in a refinery is any *decision* more important than the *activities* of locking out and tagging the equipment before mechanics begin their work on it? Completion of the checklist of activities (not decisions) that are involved in "lock-out/tag-out" is a supremely important activity to track and govern.

Another difficulty I have with decision diamonds in the human world is that, because human-level activities are typically not defined at an atomic task level, it often is debatable whether a process step is a decision or an activity. In most cases, it can be framed either way. For example, when hiring a new employee, we could frame a step in the process as "Calculate a salary based on competitive data and company guidelines." Furthermore, if we wanted to, we could add a decision diamond after that activity which asks, "Does this salary seem appropriate?" Even if that is added to our diagram, does it add any human value? When we calculated the salary, would we not naturally be asking ourselves whether the salary seemed appropriate? After all, we are not machines. We can hold many concepts in our minds at once, and then involve the right people if something seems amiss. In fact, human-level activities often involve many implicit decisions because we are applying judgment constantly (unlike machines).

That is the argument against any special value associated with the visual identification of decisions versus any other activity. Plus, even if decisions are considered particularly important to highlight for some process, there are plenty of other more economical ways to do so—for example, using a different color to shade an activity box that is a decision, adding a "D" in the corner of the activity box to designate it as a decision, or any other creative approach that is agreed as a standard.

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

Now, having deflated the one potential advantage of decision diamonds, I would like to address the disadvantages of decision diamonds. I will offer just a few bullet points:

- Decision diamonds, like swimlanes (though, admittedly not as egregiously), take up valuable process map space. This will be familiar to anyone who has used them in process models. Even in the simple *New Car Sales* process, we see that the BPMN diagram requires 9 total activities and decision diamonds (5+4), whereas the UPN diagram accomplishes a similar level of human understanding in 7 activities with no diamonds. Furthermore, rather than having floating questions beneath the diamond (partially obscured by flow lines), if a modeler decides to put the question inside the diamond, the diamond dimensions must be expanded to fit anything more than a couple of words.
- Decision diamonds are binary if usage follows BPMN standards. However, what about situations where there is a choice of several options? In a BPMN diagram, this is inelegantly handled as a series of one-at-a-time binary decisions. In other words, the desired outcome is arrived at through something that feels more like the child's game "Twenty Questions" rather than just a straightforward "human question" that enables a person to take into account many ideas at once—which is what humans are good at. Following BPMN rules, much more space is going to be consumed in a process with complex decisions, and the logic will seem terribly reductionist to humans.
- Decision diamonds introduce more mechanical complexity to the diagramming process. First, there is the issue of deciding whether to use a decision diamond or an activity box. This is usually obvious at the machine level, but at a human level some situations can seem arbitrary as to whether a decision is being made or an action is being taken. In fact, even having the decision diamond can lead to its overuse and consequent reductionism of abstract concepts, forcing a level of detail that is unnecessary and confusing. On the other hand, with UPN an activity box is the only choice for moving things forward; if the activity feels like a decision, then the modeler can start the activity with the word "Decide." As understanding about the process deepens among the team of subject matter experts and the modeler who is creating the model, it will be easy for the team to re-think the verb whenever a new insight occurs. In other words, the reduced structure and transparency of human intent frees the mind for easier innovation. Also, when people are being trained in a process, does it not seem more effective to speak to them consistently in terms of *what they do* (activity statements are always in verb-noun format), rather than also throwing questions at them from time to time? By having an activity box assigned to a role which says, "Decide...," the performer will clearly understand that it is her responsibility to make that decision. Does a floating question under a decision diamond convey the intent as clearly? Second, because there are multiple types of decision diamonds in BPMN, everyone involved in modeling or collaboration needs to understand the different meanings of these types. On the other hand, the use of words in UPN to clarify the nature of a decision is much friendlier than expecting people to remember the meaning of symbol variations.

In summary, swimlanes and decision diamonds are largely unnecessary for human understanding, they occupy valuable space on a page, and they add unnecessary complications regarding symbol usage and notational protocols.

The most common and obvious structural differences between BPMN and UPN have now been addressed. Proceeding with our exhaustive comparison, note the numbers on the activity boxes in UPN. The primary reason to display these numbers (which are optional) is to make it easier for people to refer to them when they are talking about their processes. People can say things like, "On box 4, I think the Resource should be the Sales Manager," rather than saying "On the box which says 'Order car from

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

factory,' I think the Resource should be the Sales Manager." There is some rough *human* logic to the numbering, but it does not have any precise meaning. On the other hand, in BPMN practice, activity boxes are not numbered.

It is important to ask the crucial question: "In light of its greater complexity and precision, what more information is conveyed in the BPMN diagram versus the UPN diagram?"

Actually, there are probably two answers to this question. From a machine perspective, BPMN conveys more information because the arcane symbols are recognized as part of a semantic rule set that can be interpreted by machines. But, from a human perspective—a perspective which can interpret unstructured text—the UPN diagram conveys more information!

How can this be true? There are two reasons. First, the more liberal use of words enables better understanding. Second, and even more interesting, the requirement for symbolic precision that is characteristic of BPMN *actually obscures the meaning of the process*. As an example, in the *New Car Sales* process did you notice the flaw in the process logic associated with my translation of Silver's BPMN diagram into UPN (Figure 2)? The flaw is the relationship of the credit check—box 3 in the UPN diagram—to the other activities in the diagram. As Silver's diagram seems to show (Figure 1), a Sales Rep will order a car from the factory (if it is not on the lot), the car will be shipped, then it will be prepped, and only then will we get the information about whether financing is available!

What a potentially wasteful process! All kinds of unnecessary activities are being performed in the case where financing is not available. Of course, Silver did not intend this. In fact, he terminates the process when there is no financing, *but he does not show what the implications of a termination are to the other steps in the process*. Given the lack of clarity, I had to make some guesses in a way that others might do. My version certainly seems like one possible result that would not be logically inconsistent with Silver's process, even though it would be unwise in practice. Let me take this a bit further.

The financing decision in Silver's BPMN diagram actually shows a "*financing unavailable*" result that causes the termination of the entire process. The symbol that looks like a bullseye is called a "*Terminate end event*" which causes the entire process to stop even if other parts of the process are active. Even if we know what the Terminate sign means, we have no idea what could happen to the other parts of the process since no communications to those other steps are shown. Furthermore, the concept of when things happen relative to each other is very loose in Silver's diagram. For example, if in the real world the lending agent felt ill after lunch and went home, the process seems to leave open the possibility that the factory already would have shipped the car from its own inventory to the lot only to later find out a loan was not available for the customer. Wouldn't it be better to only authorize the factory order *after* there was confirmation that credit was OK?

Therefore, the UPN diagram in Figure 2 represents an easily made human (mis)interpretation of a rather complex and incomplete BPMN diagram. Since the "*financing unavailable*" termination does not specify what happens at termination, I had to make an assumption about what would happen, so I chose the worst case (which just might happen in the real world if the Sales person only finds out about the

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

financing problem *after* the order has been filled by the factory). While the problem with the process is not very evident in the BPMN diagram, it jumps out at us in the UPN diagram. We see the output of box 3 being routed around the rest of the activities over to the closing activity. In practice, we would immediately question such a design and rework the diagram so the Sales Rep is informed before he or she places orders with the factory (or at least before the factory ships the order). A re-worked flow is shown in Figure 3—different from both Silver’s BPMN diagram and the original UPN diagram. It is likely to be more effective in teaching people what to do than either Silver’s diagram or my *translated* UPN diagram.

Note how in this improved process the flows are even more streamlined. Also, there are even more words providing context and slightly more complex instructions (though certainly not too complex for a human to understand). By using logic that works perfectly well for a human, we can have three possible outcomes from box 3: an output shown at the top left where we find out there is no financing available (because humans will understand it, we even cheated a little by stating the obvious action in the outcome rather than putting in another activity box), and two possible outcomes where there is financing, one on the top right where the car is on the dealer’s lot, and the other on the immediate right where the car needs to be ordered from the factory. Because we can rely on humans to understand this situation pretty easily, we do not need to worry about confusion over these complex, non-binary choices that would take at least a couple binary decision diamonds to describe in a BPMN diagram.

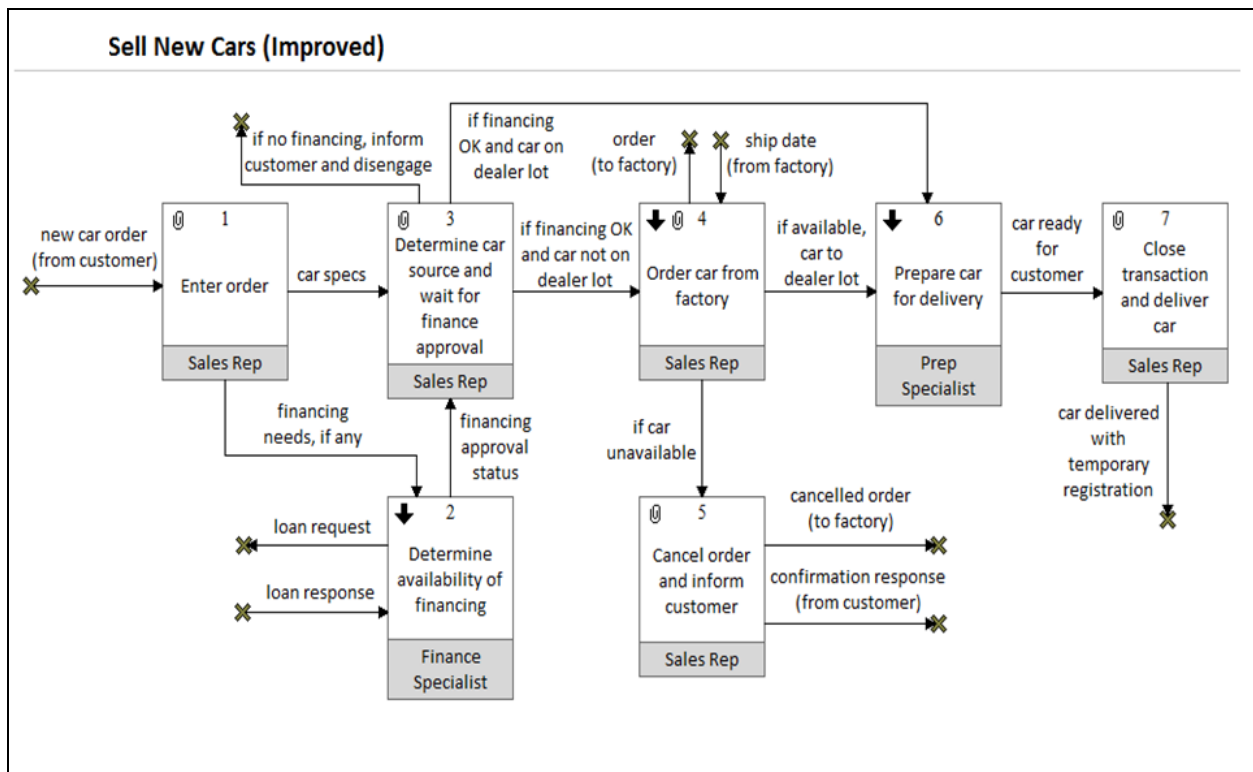


Figure 3: Improved UPN representation of a *Sell New Cars* process



## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

Also note that in Figure 3 I have replaced some of the decomposition arrows with paper clips (representing attachments). Instead of a decomposition to lower level activities (also known as a *drilldown*), by clicking on the paper clip the Sales Rep can, for example, launch from box 1 a sales order form which automatically sends data to the Finance Specialist (invoking, in a separate but linked graphical process layer, a computer-based action that could be specified in BPMN at the level of detail at which BPMN works best). Likewise, in box 3 the Sales Rep might check the paper clip for access to an inventory system that shows whether the desired car is on the lot. Lastly, on box 7 the paper clip may contain a printable checklist for use during the customer closing step. In applications like Nimbus' Control, the paper clips and drilldowns include the option of a mouse-rollover hint. In other words, the process space becomes a canvass upon which is attached a lot of information that is useful to humans. In fact, when process models were built in Control at Chevron, we found that it is often good to ask whether there is anything electronic that might be useful to the process performer that has not yet been attached to the process. Not only can the attachments be used for execution support, they serve as reportable references when a process and its artifacts are being analyzed or studied.

Frankly, it is difficult to find any information on the BPMN diagram that would be useful to a human that is not conveyed on the UPN diagram in a much easier to understand format. Even the message flows are shown, and because of the much simpler structure of UPN, it is easier to follow the process flows.

To further reinforce the implications of this point, note that we have been addressing a quite simple process. For more complex processes, the challenges of complying with BPMN structure and symbolic precision are multiplied. I have produced my share of fairly complex UPN diagrams as well, but at least the concepts in UPN diagrams are conveyed as simply as possible. As Albert Einstein said, *"Everything should be made as simple as possible, but not simpler."* To add unnecessary structural and symbolic complexity is heading the wrong direction.

Needless to say, I have pushed the case for a specific kind of notational standard for descriptive process modeling that happens to contradict the prevailing opinion of some BPM experts. This is an issue that will need to be decided in your organization. If possible, test out both approaches for a few weeks. Compare them on the following parameters:

- Ease of learning for modelers
- Ease of learning for business people (who are the subject matter experts for what they do)
- Depth of insight for business people
- Ease of direct capture (i.e., where models are built directly into the process repository environment rather than first captured offline)
- Sense of pride and ownership in process models by business people
- Value of the resultant process models to an experienced BPMN modeler to create analytical BPMN diagrams from which executable code can be developed. This is the one parameter where descriptive BPMN *might* score higher than UPN—more likely for fairly simple, straight-through processing with a minimum number of exception paths. However, UPN also may offer advantages over BPMN for translation to a fully described *"analytical BPMN"* diagram—because the less structured nature of UPN facilitates inclusion of context and reference materials that may provide value to analysts beyond a crudely structured BPMN

## Additional Arguments in Favor of UPN over BPMN as a Descriptive Notation

diagram which does not fully capture exception paths. (As we saw in Silver's BPMN diagram, he did not even deal with what happens in the rest of the process if financing is not available.)

Be forewarned that it is almost certain there will be employees or consultants working in your organization who will argue forcefully for use of BPMN as a descriptive process modeling notation. They may describe UPN as unsophisticated, imprecise, simplistic, retrograde, or not rigorous. I would disagree with these characterizations. However, the real issue is, who is the notation for? It is fine if analysts wish to translate UPN models into BPMN models for the purpose of analysis or simulation. But they should not impose their graphical rigidity on everyone who wants to talk about processes.

Why have I spent so much time on this one issue? The notation you choose is a foundational decision for your organization, and may play an important role in determining whether business people ever really *own* processes. Without business ownership, achieving the full potential of BPM will be difficult. Automations may propagate, and Lean Six Sigma efforts may return billions of dollars. But only with business ownership of processes will the organization be systematically self-reflective about the value that its activities produce. It is in an organization with an acute awareness of process dynamics that strategic insights and innovation occur—and continue to occur in a sustainable way. Having a language that everybody easily can become fluent in is essential for ownership, self-reflection, and broad-based, sustainable process innovation. It is my contention that UPN offers the best option as such a language.

I applaud Bruce Silver's efforts to promote a common standard for analytical notation that can be turned into executable code. I suspect he and others will disagree with my position that people are ill-advised to rely on BPMN for descriptive modeling. I will look forward to a productive debate that may lead to more insights for all practitioners.

### References:

1. Bruce Silver, *BPMN Method & Style*, Cody-Cassidy Press, 2009