Modeling Standards and Diagramming

Diagramming Patterns

Layout and Readability

There is clearly no right or wrong answer here. You need to explore the issue with the client.

Ideally guidelines will have been drawn up early in a project and codified into Modeling Standards.

If a model is fairly mature and contains 100s of badly-drawn diagrams, clearly it will not be practical to start suggesting everyone is redrawn now, so in those circumstances it might just be a few tweaks.

Some patterns I often recommend are:

Use the Align functionality to position objects on a grid (rows and columns) way

- Avoid too much empty space
- Avoid overlapping objects (except for nesting\ swim lanes)
- Avoid, as far as possible, connectors crossing over; if it is unavoidable, turn on Line Arches (Tools > Options)
- Use Bent lines rather than Straight
- Avoid having objects of different sizes
- If using Connector sets, always annotate the connectors
- Design for published output so think about how readable it will be on workstation

The last recommendation often means that I will draw a diagram so that "slaloms", i.e. the flow turns back on itself so that the whole process can fit on one page (or is visible without scrolling on a computer monitor).

This article, however, argues against doing that (and provides some useful general design patterns) - so, as I say, it is a negotiation here between you and the client as to what is appropriate:

http://www.irmuk.co.uk/articles/Business_Process_Modelling_BPMN.pdf

See also *Keeping your Process Models "Human-readable":*

http://www.bptrends.com/publicationfiles/02-01-11-COL-A-Practitioner%27s-Perspective-Models%20not%20Modules-Sharp.pdf

Colour and Font

Be aware of colour-contrast as well as actual colours - e.g. people with colour blindness may not be able to distinguish between red and green, so it is also useful to have colours that contrast when viewed in black and white and to use additional cues (e.g. red plus round edged border and green plus square edged border).

There is a whole science of colour, which cannot be covered here. In practical terms, however, diagrams should use corporate standards and you should be able to tell if a diagram is so garish it simply cannot be read easily.

These are usually fixed by changing the style or applying new styles in the Diagram template.

"Chart Junk"

There is clearly no right or wrong answer here. You need to explore the issue with the client. However, a few common patterns I have noticed, which I think should be avoided:

- Diagram Titles (redundant in published output, but sometimes necessary of the intention is to print the diagram)

- Graphics and clip art, which can make the diagram jokey (a good example of what not to do is the diagram "Process Order (layered for presentation)" in the Casewise Framework
- Multiple icons used in Shape Regions
- Diagram Key - again used throughout the Casewise Framework. For me, it "gets in the way". From a publishing perspective, a diagram key is better produced outside the model and linked into the published output. However, against this, they are useful for printing.

NOTE also: if you do have a key on a diagram, use a Help object; do not use a Process object to represent a process because them it becomes part of the catalogue of processes.

Notation and Methods

Obviously, the depth of this check depends on the methods and techniques knowledge you have outside of just Casewise.

Also, many clients want a notation that works for them (e.g. deliberately not using Connector Sets, or using the Process object with a category "Decision" to represent a Decision diamond. Thus, any proposed changes have to be discussed with the client.

A few common patterns to look out for, however:

- BPMN will usually not simulate (if this is a requirement)
- Accuracy of connector logic (does lax use of this introduce ambiguity into a diagram)?
- Misuse of Internal Events\Results (e.g. either as inputs\outputs or not understanding how they are the same object type that can be sued to link diagrams together)
- Lack of meta-model understanding so the information on a diagram is not properly categorized.

This can result in the process object being misused so that it shows logic, results, actors, and data - for example, I have seen a diagram something like this (each object being a process):

> *Customer sends an order > If it's Ok, do X > Order data is sent the Order tracking database > Order continues*

In the field of Entity Relationship Modeling, users may not be familiar with the specific types of notation used and have drawn up a hybrid diagram using Connectors instead of Relationships. If this "does the job" for the client, that's fine; but it will lack rigour and could be done in a better way.

Templates

If a client starts with a Casewise-supplied template such as the Casewise Framework or the TOGAF extension, then they will start many Diagram templates (e.g. TOGAF has around 50 plus Matrix templates while the Casewise Framework has around 25). Often, few (if any) of these will be actively used by the client, causing unnecessary confusion.

For example, when you create a new Diagram, the user is presented with a list of existing Templates. If this list contains 20 or more Templates to choose from, it increases the scope for choosing the wrong one (which I have seen in a client using TOGAF).

Essentially, every Template should be described in a Modeling Standards document and if necessary, this document - and the modeling standards - will end up being simplified. At one client, for example, they could not explain the difference between a "Business Process Diagram" and an "End-to-End Process Diagram". What transpired is that different members of the Team were using these because their preferred option was meaningful to them. In terms of actual usage, they were actually trying to do the same thing (and indeed, a "Business Process" is arguably always cross-functional and so is always "end-to-end").