Experimental features

Experimental features are the features that are still under the development phase and not ready to be deployed into the actual software or product. It is a feature where the design is not finalized.

Rather than postponing them until subsequent releases of our software, we choose to include them in our software, but disable them by default (for production environments), while easily allowing you, the user, to enable them in experimental environments, play around with them, and get back to use with feedback about your experiences.

Optional tabs

Pathomation's user interface is ribbon-based, so one way we hide experimental features from you is by commenting out sections of the ribbon definition. When you remove these comments, the new options appear automatically after refreshing your web-browser.

Pre-set annotations

This one is not as much experimental, as it is a tab that you _do_ wish to customize for your own organization's needs. After playing with PMA.studio (and particularly its annotation features) for a while, it's common to develop a routine to annotate the same kind of slide annotations again and again. Examples include:

- I want to indicate necrotic regions on my slide, so I'm going to use a black line color, with a gray fill of % opacity
- I want to indicate circular areas with a 5mm diameter to indicate the most relevant areas for tissue microarray (TMA) preparation.

Conferencing

Experienced digital pathology users are aware that slide sharing through standard video conferencing tools like Zoom or WebEx is far from ideal.

PMA.studio can interface with both PMA.core and PMA.live to provide a better way to share slide-centric discussions.

Admittedly, we got into a bit of feature creep when designing this one, and it's not quite finished yet for prime time. Experimentally, though, it's a great resource already for team discussions, and we already hear from several customers that were pretty excited playing around with initial implementations.

So, rather than putting this one in the freezer, we're simply allowing you to enable it when you want to and play around with it. Then, let us know if this is (already) useful to you, and what edge conditions you'd like to have more control over.

Artificial intelligence (AI)

There's no question that everything artificial intelligence is smoking hot right now. All the big guys are doing it, so why shouldn't we?

Summarizing the above blog article; Pathomation doesn't want to come out with its own AI software, but it does want to offer a mechanism that allows data from different AI providers to come together in a single uniform interface.

This is where the custom panels come into play.

PMA.studio comes with an option AI ribbon tab that repeats relevant annotation capabilities from the annotation tab, and provides interfaces to a number of well-known AI providers already.

But better yet: the interfaces that we offer this way are implemented through custom panels (following the "take your own medicine" principle). The code behind these is currently written in PHP, and can be used for study and to build your own interface to your own AI providers (or further extend the ones we offer (limited) support for already ourselves.

Persistent user features

While the panels and ribbon tabs in PMA.studio are fully configurable, they have a major flaw: in their current implementation they're only configurable at the application level.

Furthermore, a user must have PMA.core administrator-level access to be able to modify either configuration. This last one is actually not a bad thing, considering:

- Both panel and ribbon configurations require knowledge of XML, and
- Changes in the configuration will affect *all* users of said PMA.studio instance

The solution is to have a basic configuration that applies to all users, and in addition allow users to define their own interfaces as they see fit. Other applications like Microsoft Office 365 have had this feature for a while now, of course, but then again, Office users historically only had to worry about their own configuration on their own computer. In our case, we need to thread a bit more carefully.

The key word is "user persistence".

We've started with making some basic features persistent: end-user settings like "auto-save annotations" and others, all accessible from the Edit settings from the application's context-menu.

In future releases of PMA.studio you can expect to see a lot more persistance.

See also

Tab inventory

Available ribbon commands