iFISH - Conceptually

What is iFISH?

iFISH is an underlying technology that can form the basis of discrete online applications designed to help users explore information in an engaging, enjoyable and effective manner. It provides users with a unique exploration experience. iFISH offers a playful environment designed to encourage users to explore a wide range of items based on their personal preferences and discover new items that they might otherwise not have discovered.

We regard 'exploration', or 'browsing', as distinct from online 'search' in that the user often has a poor idea of what they are looking for and hence is not inclined to want to use keywords in a traditional search environment. An exploration experience needs to be fluid, be easy to retrace steps, and present potential 'hits' in a strongly visual manner that encourages a further quick and deeper investigation.

iFISH provides all of the above. It employs sliders to rank explored items using fuzzy logic to match the user's preferences. But it also has some unique features designed to improve the user's experience and sustain curiosity during the process. These are discussed below.

See examples of iFISH in action: http://bookExploramatic.com and http://tiny.cc/cisfish

What does iFISH do differently?

Apart from a highly interactive and evocative experience, a typical iFISH application has five unique features that distinguish it from a traditional search engine:

1. A **small number of sliders** drive users' explorations providing fast, animated responses.
   In contrast, a typical search engine requires text input, RETURN, read list, text input..., etc.
2. A **smart text-input search** feature allows users to easily find an item that they already know about. The system adjusts the sliders to 'float' this item to the top of the display, hence bringing other similarly tagged items with it. This a powerful way to find new items that have similar personal appeal to the original one.
   Some traditional search engines use a 'recommender system' approach to do this, proffering items to the user based on a propriety algorithm.
3. A **'popularity' slider** lets the user gradually reorder items by using slider that takes into account the 'thumbs up' vote of previous users.
   Traditional search engines might indicate popularity but not allow the user to do anything with this information other than a simple sort.
4. The user can **view how an item has been 'tagged'** and, if maybe in disagreement, add their own tag values to the mix. This creates a system that dynamically modifies its behaviour based on crowd-sourced data.
   Traditional search engines tend to have fixed categorisation data associated with each item.
5. A slider, ranging **from 'Expert' to 'Community'**, lets the user explore the impact of the community's tag inputs over that of the original 'expert'. It lets users choose by how much they wish to see a *system* view of the data, or a *community* view of the data.
   The traditional search engines have a different approach to this issue by making suggestions based on "People who like this also liked...".

Other features of an iFISH application:

6. A set of filters can be used to reduce the scope of the items being explored.
7. Users can disable sliders that are of no personal relevance.
8. 'Favourite' items can be dragged aside into a 'save list'.
9. Clicking found items presents a more detailed view that can be used to click through to further web pages, sales information, etc.

Variations in iFISH applications

Just what an iFISH application looks like depends on its customisation for a particular purpose. It could be created as a 'web app' and run on all mobile and desktop platforms via web browsers, or it might be developed as a native app for iOS, Android or other mobile devices.

Managing fish ponds

Yes, we use a very fishy metaphor for the iFISH environment! We refer to the data as "fish" and the structure of a particular iFISH instance as a "pond". We have a very sophisticated web-based backend that makes it easy to manage all aspects of a pond: add & edit fish, define & create new ponds, see overviews of all pond data, manage users.

And why "iFISH"?

That's a long story, but the crux of it is that our first dabble in this space was to create a system for students to explore subjects within the university's handbook. We called it **i**nteractive **F**oraging **I**n the **S**ubject **H**andbook. The concept of "fishing" for information worked well – the name stuck!

More fishy examples can be found at http://people.eng.unimelb.edu.au/jonmp/projects/iFISH/

Jon Pearce

August 2013