

comp.lang.python: Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004-09/4669.html>

From: Richard Hanson (*me_at_privacy.net*)

Date: 09/25/04

Date: Sat, 25 Sep 2004 12:37:50 -0700

Alex Martelli wrote:

> Richard Hanson <me@privacy.net> wrote:
> ...
> > > search facilities _together_ with good organization of your materials
> > > (if you take the bother of the latter). Consider Mail.app: its search
> > > functionality works across all mailboxes or on a single mailbox --
> > > you're _still_ encouraged to do a little decent filing of your mails,
> >
> > It *does** seem necessary to "educate" the database about our own
> > preferences and styles of organization and the like, as well as
> > "rating and filing" specific, individual objects. I'm interested in
> > figuring out ways to make it *very** easy to "add value" to the
> > database with a more efficient HCI -- unless it's very, very simple to
> > do (nearly automatic, even ; -), the user won't do it.
>
> There are many kinds of users.

Agreed -- I spoke too loosely, there.

> I essentially rely on automatic
> classification of mails into mailboxes based on a dozen or so
> rules/filters. But my wife Anna, who's worked as office manager etc for
> years, has filing as "second nature" -- _her_ mailboxes (Thunderbird,
> her preference) are neat and pristine, just like her filing cabinets.
>
> In real life whenever I have to look for some weird document I face
> hours of rummaging through papers trying to divine where I may have put
> it; when Anna is looking for a document in her archives, she reaches
> into the right folder, and there it is, period. On computers, with
> automatic classification rules and search engines &c, her advantage is
> not quite as pronounced -- but it's still there.
>
> I have known several people with a penchant for effective and systematic
> filing, though they are no doubt a minority, wrt us slobs. It's
> important for the system not to get in their way: for example, when they

Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

comp.lang.python: Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

- > *move a mail from folder 'pending purchase decisions' to folder*
- > *'purchases considered but rejected', the system must automatically and*
- > *seamlessly update the indexes of both folders, and the global index of*
- > *all boxes, too. Otherwise, if the system makes it _less_ effective to*
- > *do good classification and filing, it will earn deserved scorn and*
- > *enmity from those kind of users.*

One problem I have with tree organizations is that often I want an object to be in several or many branches of the tree -- the memory cost for larger objects, and the synchronizational problems resulting from the replication of such, can become problematic. I like the flat-file approach with "virtual folders," if you will, through the definition and addition of "folder attributes" (again, if you will) attached to the individual objects. This approach makes possible the ability to have multiple virtual trees of organization with easy switching from one organization to another. (Possibly, a *partial* emulation of this part of my idea could be implemented using Linux and Unix's "sym-links" [if I have that term correct]. However, I would prefer a proof-of-concept work on top of Windows, as well, and therefore I'm thinking of a database with one field containing the object's path, leaving the objects (for now) in the hierarchical harddrive's tree.)

- > > > *though the search does make it more feasible to survive with the popular*
- > > > *"one big inbox and never bother filing" paradigm;-).*
- > >
- > > *I'm thinking one database with some auto-indexing helped along with*
- > > *specific user guidance re the "value-adding" aspect. Without some sort*
- > > *of fast auto-indexing, of course, and without the easy ability for*
- > > *user attribute-adding, the "one big box" paradigm could be painfully*
- > > *slow for precise narrowing of large collections of heterogeneous*
- > > *objects. Presumably, the above mentioned apps, and Google, of course,*
- > > *do the auto-indexing part.*
- >
- > *Yes, and 'searchlight', Tiger's forthcoming search engine, centralizes*
- > *the indexing (still needs SOME cooperation from apps, of course) as well*
- > *as the search. The ability for the user to tag documents with arbitrary*
- > *metadata is also a key part of this, I assume it's what you call the*
- > *"value adding aspect"?*

Correct -- that is what I mean. Also, a key point is that it needs to be drop-dead easy to "add value" (the arbitrary metadata) so that lazy humans like me will do such. :-)

- > > *Apple's work definitely sounds like a step in the right direction. Is*
- > > *there *complete* integration of *all* object types?*
- >
- > *Only in as much as applications cooperate with the central engine, at*
- > *least in some aspects. That may be why Apple has been circulating*
- > *alphas of Tiger to developers with such HUGE lead time, up to a year*
- > *before it hits the shelves: Searchlight's value grows exponentially the*

Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

comp.lang.python: Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

- > *more apps register their documents' metadata with it, and use it for*
- > *their search facilities -- app developers need time to enable that. Say*
- > *a user prefers Entourage or Thunderbird to Mail.app for their mails:*
- > *such a user will find Searchlight less useful than a Mail.app user will*
- > *_unless_ these other mail apps cooperate with Searchlight (sure, Apple*
- > *might reverse engineer _some_ file formats for documents of some*
- > *recalcitrant apps of particular importance, but that's more overall*
- > *work, and will still tend to produce results not quite as satisfactory,*
- > *than if the app's own developers did the job...).*

Again, this is sounding quite encouraging -- on Apple's part, at least.

- >> *As I see it, the user should be able to tap a button and change an*
- >> *email into a snailmail doc, or vice versa, say, with the system*
- >> *handling the details automagically.*
- >
- > *In Tiger, such import/export facilities remain fully up to individual*
- > *apps -- there's just no way an indexing and search facility can have*
- > *enough metadata on all aspects of documents of disparate types*
- > *(including, crucially, formatting/presentation issues) to do the job.*

Check. That's why I think an "appliance" or embedded approach is a nice target. *One* development team makes the one, monolithic "app" with the OS *and* user functionality integrated within.

- >> *And, with a few appropriately set*
- >> *filters using attributes, sorts, and such, the user should be able to*
- >> *see all the emails, documents, pics, sounds, flicks, etc. relating to,*
- >> *for example, "Pink Floyd" -- all in one, narrowed view even though the*
- >> *objects are of disparate types.*
- >
- > *Yes, this IS part of Searchlight's functionality, and part of the added*
- > *value of a centralized indexing/search wrt per-app facilities.*

Again, quite encouraging.

One of the reasons I started this thread was to tease out other work related to mine which I was unaware of. I have long observed that many folks independently but concurrently come up with the same epiphanies and ideas. I had sincerely hoped that this was the case with my ideas, as well. (Indeed, I was presuming such -- not wanting to reinvent the wheel, and realizing that many, much-more-capable people are on this planet. :-))

- >> *I don't have access to the modern Macs -- they sound more interesting*
- >> *to me than the Wintels I'm using. (Although, pedantically speaking, my*
- >> *currently not-working Fujitsu laptop uses a Transmeta chip, not an*
- >> *Intel one.)*
- >
- > *Heh -- Anna's got a Fujitsu with a Transmeta chip, too (P2000 Lifebook,*

Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

comp.lang.python: Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

- > *works fine, alas not very fine with Linux yet;-). And all the Linux*
- > *boxes in the house have AMD chips, better value-for-money (there _is_ an*
- > *OpenBSD box with a pentium 75 that does routing, firewalling &c year*
- > *after year since the dark ages, admittedly).*

Mine is a P1120 LifeBook with touchscreen. I have read about two people who were successful in getting Linux running with the touchscreen *somewhat* operational on the P1120. The touchscreen is particularly important to me for the several reasons I mentioned previously. (I plan on getting a new harddrive for the P1120, soon, and then see if I can get a Linux distro installed with the touchscreen fully functional.)

- > *Today's Macs don't have Tiger yet (unless an alpha-stage developer*
- > *preview), though some of the (mostly per-app) search facilities are*
- > *indeed excellent even in Panther, the current release of MacOS X. Tiger*
- > *should be out in the spring of 2005, and it will cost about \$150 (free*
- > *with _new_ Macs, but \$150 to upgrade existing ones).*

Indeed, it sounds *very* interesting.

- > > *Mike Meyer's link to an implementation of Jeff Raskin's work also*
- > > *sounds quite interesting. I note from scanning the smaller zipfile*
- > > *referenced in the link some similarity with my own ideas -- no save or*
- > > *delete, for example, such being either unnecessary or is transparently*
- > > *handled under-the-covers.*
- >
- > *Yes, Raskin's paradigms are definitely revolutionary, for good or for*
- > *evil.*

;-)

- > *Apple (and I guess MS for Longhorn) OTOH are living in the real*
- > *world, where apps from multiple suppliers with their own separate*
- > *documents (sometimes in proprietary formats, sigh) are a reality and*
- > *users won't drop them (particularly as they need to keep exchanging docs*
- > *with other people using other machines), so any paradigm shift must*
- > *accomodate the likelihood of gradual and incomplete transitions...*

I understand your point about Apple et al starting from the current real-world situation and mapping out a route forward, starting from here. That's a necessary step in the evolutionary process -- I totally concur. My project aims for a more distant, down-the-road point, with parts of the project not even solvable at this time (the hardware platform, for instance), and without a transitional route mapped out. Thus, my desire to realize a prototype on top of the current platforms as a proof-of-concept. (We'll see if I have the energy and wherewithal to get anywhere. :-))

Thanks for the enlightening comments!

Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

comp.lang.python: Re: [OT] "Pre-announcement" of Python-based "computing appliance" project.

life-is-all-about-taxonomy'ly y'rs
Richard Hanson

--

sick<PERIOD>old<POINT>fart<PIE-DEC0-SYMBOL>newsguy<MARK>com