What scares you the most about the integrated IDE of most modern Smalltalks? [closed]

Asked 16 years, 2 months ago Modified 3 years, 5 months ago

Viewed 2k times

22

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 13 years ago.

As I'm riding the wave of resurgence of Smalltalk (especially because many Ruby-on-Rails people are rediscovering Smalltalk and seeing Seaside as their next upgraded web framework), I get questions like "yeah, but how do I use my favorite editor to edit Smalltalk code?" or "Does Smalltalk still insist on living in a world of its own?".

Now, having first experienced Smalltalk back in 1981, I don't understand these questions very well. It seems rather natural that I'd want the editor and debugger to be savvy of my current code state, and integrate with the change control system that is Smalltalk-aware. Using an external editor or debugger or change control manager would seem very awkward.

So what is it that scares you the most about not being able to edit the five-line methods in Smalltalk with your favorite editor, or use your favorite non-Smalltalk-aware change control system?

ruby-on-rails ruby ide smalltalk seaside

Share

Improve this question

Follow

asked Oct 7, 2008 at 16:13

Randal Schwartz
43.8k ● 4 ● 49 ● 82

8 Answers

Sorted by: Highest score (default)

27

Everything's different. Want to go to the end of the line? It's not Ctrl - E. Want to jump a few words over, by word? It's not Meta-F....

Text editing is a *fundamental programming activity*. Messing with those inputs is messing with something deep in my mind.

Edit: and here is someone asking for emacs key bindings on comp.lang.smalltalk in *1987*.

Share Improve this answer

edited Jun 27, 2021 at 14:16

Follow

AEM
1,386 ● 8 ● 24 ● 33

answered Oct 7, 2008 at 16:30

EfForEffort
55.9k ● 5 ● 38 ● 41

Having said that, It shouldn't be that hard to retrofit emacs-isms to the editor in the VM. By the very nature of Smalltalk you've got the source code for the class browser and editor. – ConcernedOfTunbridgeWells Dec 15, 2008 at 18:44

So in other words, you're afraid of change. – Dog Apr 9, 2013 at 13:33

Just because it can be fixed doesn't mean you should expect people to waste their time overcoming such barriers to entry. It should just work the way they expect it. Image you go into a car and it has no steering wheel. Are you afraid of change? – Alexander Feb 19, 2016 at 5:54

▲

10

▼

The only Smalltalk I've spent any time with is Squeak, so my views may not apply to other Smalltalk environments.

What concerns me about the image-based approach is that, while you have wonderful things in the Smalltalk environment, it is a walled garden that makes it difficult to interoperate with anything outside that environment. For example, what if I want to use external tools like Yacc and Lex? What if I want to use some C or Python programs to generate Smalltalk code? What if I want to mix Smalltalk in with a bunch of code written in other languages, editing code in all those languages in one editor and keeping it all stored in the same source-code tree?

I'm sure it's possible to deal with all these issues by having your Smalltalk environment invoke system functions to control external tools. But how easy is it to let external tools control your Smalltalk environment? In other words, what if I want Smalltalk to be just another component, rather than the master of everything?

Share Improve this answer Follow

answered Oct 10, 2008 at 15:15

community wiki
Kristopher Johnson

1 This is still a key weakness which keeps me from considering Squeak/Pharo for serious work. – Warren P May 26, 2010 at 19:21

"What if I want to mix Smalltalk in with a bunch of code written in other languages ...": Alex Warth's Ometa/Squeak does exactly this: you write some "methods" in Ometa, and others in Smalltalk. It's still a bit rough, but the idea shows that interoperating's certainly possible. See also Renggli's Helvetia work: scg.unibe.ch/archive/papers/Reng09bLanguageShootout.pdf – Frank Shearar Sep 21, 2010 at 11:35

9

Nothing scares me in particular, but I found working out the API's in VW a bit of a chore, even when I had used other smalltalks. The effect of the browsers is that you tend to see the API's a little bit at a time and quite often it's not immediately obvious where you should look for particular functionality.

Smalltalk also suffers a bit from the paradigm shift to understand how it works. When I was doing my bachelor's degree at university (some time after I had first encountered Smalltalk) I got to enjoy a bit of Schadenfraude watching everyone else in the class getting over the initial paradigm hump as they learned the system (Squeak) for the first time.

I think the combination of the paradigm shift and functionality being somewhat buried in the class libraries makes for a bit of a steep learning curve. ST had a reputation for a fairly steep learning curve to really come up to speed - most of this is due to the large class libraries and the fact that most of the language functionality is buried somewhere in the libraries.

Also (and sadly), Java came along in the mid 1990s and grabbed all of the mindshare. The major Smalltalks have either died completely or been sold off to niche players. It's quite Ironic (in a happy way) that Ruby has served to re-awaken interest in Smalltalk but the lingering perception of 'also-ran' obsolescence doesn't help.

See This post of mine for some pontification about the merits (as I see them) of getting heavily involved in Smalltalk in this day and age.

I would be quite happy to go back into Smalltalk if the opportunity were to arise.

Share Improve this answer Follow

edited May 23, 2017 at 11:53

Community Bot
1 ● 1

answered Oct 7, 2008 at 17:31

ConcernedOfTunbridge Wells
66.5k ● 15 ● 148 ● 198

8

The one big show-stopper for me is that code I write one Smalltalk VM is STILL, after all these years, not compatible with other Smalltalk VMs.

I understand why that is: the core of Smalltalk is an extremely small set of axioms and keywords. This means that after 30 minutes of learning Smalltalk, you're already learning the API library rather than the language itself. I like that approach to language design.

What it all boils down to however, in the Smalltalk world, is that unless a consensus is reached between all VM vendors to have a common base Standard API, my Smalltalk code written for one VM is almost certain not to run on other VMs when I decide to switch.

This also has the corollary of obsoleting part of my knowledge of the space when I switch VMs.

Note that I have barely tried Smalltalk in my life. I'm far from being an expert. This understanding comes from speaking with James Robertson about a month ago.

Another point I'd like to make is that Seaside does in fact run on most popular Smalltalk VMs. I wonder how much of (what should have been) a Standard API they had to build for themselves to achieve that feat.

With all that said, I always have an ear out to hear more about the state of Smalltalk. I *do* want to try out Smalltalk's very powerful development environment (and its other goodies).

Share Improve this answer Follow

answered Oct 7, 2008 at 22:51

webmat
60.5k ● 12 ● 56 ● 59

We haven't had to re-*invent* too much. Mostly it's just limiting ourselves to a common API subset shared between Smalltalks. Then there are a few hoops to jump through and the porters still have to do a little massaging. It would certainly be nice if some of the core things were more standard. – Julian Oct 9, 2008 at 20:14

Ah, ok. Then maybe it's not as bad as I thought it was :-) It certainly doesn't look ideal, but if at least the APIs look a bit alike, then at least I don't scrap all my knowledge when moving from one platform to the other. – webmat Oct 10, 2008 at 15:58

You should see all the problems with different C, C++, etc. compilers. Not to mention that most dynamic languages tend to only have a single dominant VM so they completely ignore this problem -_-' – user9903 Feb 11, 2012 at 19:08

3

I know it's late but the biggest annoyance for me is that there is not really good editor in none of the smalltalks. It's a thing I can not understand. Working with text is so essential and that less "supported"....

It's always this just staring at one method and then you need to have some method finder or another browser around just to check another method. This is what I really dislike....

Share Improve this answer Follow

answered Mar 24, 2009 at 16:06

Friedrich

5,996 ● 1 ● 29 ● 46

You don't need a powerful editor when you're dealing with 5 to 10 lines of code at a time. The integration of the editor with the environment helps substantially too. – Randal Schwartz Mar 24, 2009 at 21:25

2 I disagree about that, and I'd prefer a Smalltalk with decent edit facilities. Just imagine a Mail program "embedded" in Squeak and you'd agree that good editors would be fine.... – Friedrich Mar 29, 2009 at 6:32

Omnibrowser has a "chasing" browser that makes finding senders-of/implementors-of MUCH easier. Look at OBImplementorsBrowser, for example. – Frank Shearar Apr 1, 2009 at 8:00

1 Randal, I agree. Friedrich, I agree too :) The editors (hell, the whole UI in all Smalltalks, except maybe Dolphin*) could benefit from some serious usability polish, even if we don't need an emacs per method. * feels funny to say that the windoze-specifig one is the most usable :(– Damien Pollet Apr 9, 2009 at 21:16

I find that I don't need a heavy-duty editor feature to play with Squeak/Pharo. And so far, all I can do in it is play. I'd feel good if I could get far enough along to be writing enough code that I miss some vi or emacs, or windows editor features. – Warren P May 26, 2010 at 19:25

1

While the restricted Smalltalk environment made things like **relying** on a database driven source control system possible at times where other languages still struggled with having a proper editor, it makes **integration** very hard in todays times.

With tools like Eclipse or Team Foundation Server you get so used to having all tools integrate with each other. E.g. if a requirement is created, it is automatically linked to the change sets that the programmer commits to implement that requirement. This "boundary breaking" between formerly different tools is nearly impossible in the Smalltalk world, but with bigger projects, bigger teams, higher levels of abstraction and so on you need tools which are more than a fancy editor and help you throughout a full software development life cycle.

Share Improve this answer Follow

answered Dec 7, 2008 at 13:50

Bananeweizen
22.1k ● 8 ● 72 ● 92

But eclipse is a bit like a Smalltalk image, no? Everything has to be inside the workspace, everything has to be done through eclipse plugins... can I git clone blah somewhere else and still have eclipse integrate with that? – Damien Pollet Apr 9, 2009 at 20:57

Also, boundary breaking *is* impossible in Smalltalk, because there simply is no boundary. Version control tools and code browsers work on the same class and method objects that the VM runs. Press ctrl-T on a test class in the code browser: a red/green dot appears, no need to run the SUnit UI, etc. – Damien Pollet Apr 9, 2009 at 21:11

1

No useful support for navigating with the keyboard, or supporting platform UI behavior.

While it's true you don't really need an incredible text editor for (well-written) Smalltalk, being able to move around the environment while keeping your hands on the keyboard is quite useful (and in my case, essential to reducing RSI). I just was trying VisualWorks' inspector and the arrow keys didn't even work properly to move up and down a list. When I hit the space bar, I got a walkback. Sigh.

Share Improve this answer Follow

answered Oct 21, 2009 at 2:44

Nicholas Riley
44.3k ● 6 ● 104 ● 125

▲

0

▼

For the Windows world, there is nothing like Dolphin Smalltalk. The IDE is fantastic. Another quality product if you want to try is Visualworks, it works well, has a very fast VM and the documentation is pretty good.

I've used both in the past, there is nothing to fear.

Share Improve this answer Follow

answered Dec 21, 2008 at 23:47

Hernán

4,587 ● 4 ● 34 ● 48