

Ken Albers: Let's start with some background information

KA: When did you begin using computers? How did you get interested in computers?

Alex Vincent: Well, I can't quite answer that question with certainty, because I literally don't remember... according to my Dad, I woke him up in the middle of the night when I was three years old to save a little program to cassette from a TI-99/4A. I'm 29 now, so you could say over twenty five years ago.

AV: I simply grew up with them.

KA: Wow, that's amazing

KA: So did you ever have formal training, or were you mostly self-taught?

AV: aside from my typing class in high school, everything about computers I've pretty much taught myself - but I have a full bookshelf of technical references.

KA: What's the first programming project you remember working on?

AV: That would be the summer after my typing class (this will take a moment to explain)

KA: ok

AV: you know the old adage about programmers wanting to scratch an itch? Well, I truly experienced that for the first time during my typing class.

AV: We had a typing drills program - basically strings of text we're supposed to retype, character for character, within a minute. It tested accuracy and speed.

AV: There was just one little problem with the program... the strings of text had spelling errors. Everywhere.

AV: For me, that was aggravating in the extreme. I take some measure of self-respect in my ability to write, especially properly, so the spelling errors problem made the program harder to use than it had to be.

AV: I talked it over with the class teachers, and arranged to borrow the source code of the program, written in BASICA or GW-BASIC - I don't remember which.

AV: -- this was Nathan Hale High School, by the way

AV: Seattle, WA, USA

KA: And so you reworked the code?

AV: now, the source code for this program was encrypted, and about a month later I found out you couldn't decrypt it.

AV: So I didn't rework it. I rewrote it, from scratch, in QBASIC.

KA: Wow.

AV: not the QuickBASIC that Microsoft was selling at the time, but the free QBASIC that came with MS-DOS 5

AV: well, as I said, it was a simple typing drills program; it stores the strings of text in its memory, and it had a simple text interface.

AV: in retrospect, I'm glad it was something that simple; I could not have done the same with anything more advanced.

AV: I had the first draft finished a little later than I hoped, I think - about a month into my junior year

AV: I don't remember when I finished it, and it took us a couple months to find the time to get it fully installed

AV: (me and the typing teachers)

KA: :)

AV: As a bonus that I didn't expect, they gave me an "A" grade on my report card for completing their programming class which I had never taken

KA: Ha-that's great!

KA: So, how did you first connect with open source projects and begin contributing to them?

AV: Actually contributing? That really depends on what you define as a contribution - bug reports, triage, or submitted patches.

AV: or something else :)

KA: I guess maybe trace your trajectory from first encounter to becoming a full Mozilla developer.

AV: Okay, first encounter. This actually wasn't with Mozilla code at all, but with a little tutorials site called Website Abstraction (now JavaScriptKit.com)

AV: basically, sometime after my discharge from the U.S. Navy (or maybe before that, I don't recall), I was tinkering around with JavaScript, which looked like an interesting language to play with. I even had bought this orange book for JavaScript beginners - I don't remember the title, but it had on the cover a well-known actor and a joke about JavaScript instead of Folgers coffee...

KA: :)

AV: I found out the hard way it really was for beginners, just an introduction to the language and not much beyond that

AV: anyway, I was playing around with JS, and I was volunteering a little with Website Abstraction, and I wrote an article for them.

AV: Then I wrote another, and then another, and then another.

AV: -- I should note that I like to push the boundaries of what's possible on the computer - I'll touch on this later, but that's how most of my articles come about... when I develop a new tool or technique that I don't see anyone else doing and I think it's useful, I try to write about it so that others can learn.

AV: anyway, I did this for a little while. Now, this was in the heyday of Netscape (which I've never worked for, nor for its offspring, the Mozilla Foundation and the Mozilla Corporation)

AV: I was a big fan of Netscape, and I preferred their browser simply because I'd been exposed to it first on Macintosh computers in the Nathan Hale computer lab.

AV: One of the big revisions of JavaScript, I think 1.5, had just arrived, but unlike JS 1.2, there was no written JavaScript 1.5 Reference or JavaScript 1.5 Guide at the time.

AV: so I popped into one of the newsgroups and asked if anyone was working on it.

AV: A Netscape employee by the name of Steve Rudman said no, there wasn't... but would I like to write it?

AV: Here I made a rather big blunder. I said "no". I said I didn't know how to do so.

KA: So, nonetheless, you were getting pretty quick feedback on your articles and queries?

AV: yes, I was. Let me pause for a moment, please, to give you a little more background on what I was doing at the time.

KA: ok

AV: All this time I'm talking about writing articles, tinkering with JavaScript, etc., was spare time. It wasn't my day job.

AV: My day job, well...

AV: bear in mind I'm right around the age of 20, and still pretty naive about the real world

KA: Ok

AV: basically, my day job was forgettable

KA: As most are when you are 20...

AV: (I don't want to really talk about it within the context of this interview, but it's pretty silly)

AV: it was a direct marketing position.

AV: part of that job entailed reading Anthony Robbins's book, "Awaken the Giant Within".

KA: :)

AV: through reading that book, eight months after I told Steve Rudman no, I realized my blunder. That I really could have written that tutorial, that I was judging myself short when someone else had thought I was capable.

KA: Interesting

AV: I jumped on the newsgroup, asking if the contract was still available, and Steve Rudman said, "Sorry, we gave it to someone else." Which then led to my first big break...

KA: Which was?

AV: http://www.amazon.com/JavaScript-Developers-Dictionary-Alexander-Vincent/dp/0672322013/ref=sr_1_1/002-0701837-1798461?ie=UTF8&s=books&qid=1176229083&sr=8-1

KA: Ah yes, the Developer's Dictionary

AV: Shelley Johnston of Sams Publishing was watching the newsgroup at the time, and she was looking for an author to write a book in their Developer's Dictionary line for JavaScript.

AV: this time, I said "YES!"

AV: it took me a lot longer than I thought - I changed jobs (from one low-paying one to another), and it ultimately took me over a year

AV: Now, I should emphasize something.

AV: Before I started on the book, I thought I knew JavaScript fairly well. I'd written a bunch of articles and several useful JS library tools.

AV: Boy, was I wrong!

AV: for this, I should credit Martin Honnen who understood the language a bit better and was my technical editor on the book, for helping to straighten me out. I also need to credit a couple guys at Netscape who were patient with me when my questions didn't make any sense (this is where the open-source angle comes in)

KA: So did you utilize the web community much in the process?

AV: up until then, I didn't really realize there was a web community - or more accurately, an open-source programming web community.

KA: And this was around 1998-99?

AV: no, this would be 2001

KA: Ok

AV: I was asking Arun Ranganathan (aruner) and I think jband (not sure about that anymore) about what DOM HTML support there was beyond what the W3C DOM specs stated.

AV: unfortunately, they didn't understand at the time, including a one-hour phone call that didn't clear up anything.

AV: Then over IRC (I think), they finally pointed me to LXR and the IDL files for the Document Object Model which mozilla.org uses.

AV: and I said to myself, "This is it! This is what I've been looking for."

AV: I wrote the second half of my book based largely on that and on Microsoft's documentation - and on my own test cases.

AV: that was my first exposure to the open-source community, as we understand it today.

AV: Arun and the other gentleman got free copies of my book, which came out at about the same time as Mozilla 1.0 (not Firefox 1.0, this predates that). My book hit the shelves in May, 2002.

KA: Fascinating.

KA: And when did you begin to contribute to Mozilla?

AV: not long after. I found out the user interface for Mozilla was written in XUL (a XML language), XBL (another XML language), Cascading Style Sheets and...

AV: JavaScript.

AV: That was sort of a "Eureka" moment for me.

KA: So you were able to hook in because of the JS?

AV: yes.

AV: the first major area of work I put into the mozilla.org project was in DOM Inspector, which Joe Hewitt created and Christopher Aillon at the time was the owner of.

AV: unfortunately, that was also when I ran head-first into the code review process.

KA: So did you begin by submitting items through bugzilla?

AV: yes

AV: now, I respect code review, I think it's essential, but it takes quite a while :)

KA: So you found it difficult to break into the community at first?

AV: I'm not sure. I already had a hand in the pot with IRC and my book.

AV: I found myself writing a bunch of patches to make what I considered essential improvements to DOM-I.

AV: and as I did so, a bunch of patches (probably 75-90% of them) got r- or sr-

AV: which caused me to rewrite the patches, only to get r- or sr- again.

AV: It's a very painful way to learn the development process

KA: I'm sorry, I'm not familiar-what do ~r and ~sr indicate?

AV: r refers to "patch review". r- means "This patch needs more work."

AV: r+ means "Yes, I think this patch is good enough to land."

AV: sr refers to what the mozilla.org community calls "super-review" - basically instead of a module-level review of a patch, a review of the patch by a highly qualified member of the community, mainly looking at the patch as it applies to the whole mozilla.org project

KA: Thanks

KA: I can imagine that would be frustrating to be volunteering your time and efforts only to need to rework your code...

AV: at one point the term "strong review" was suggested, but it never took off.

AV: Frustrating? :)

AV: Here's frustrating: https://bugzilla.mozilla.org/show_bug.cgi?id=112775

KA: lol

AV: I submitted a lot of patches on that bug (in addition to the bug itself) for something I felt DOM-I must have and should be easy to do: inserting nodes into the DOM.

AV: I kept hammering at it, hammering at it, and eventually gave up.

KA: So as a volunteer, did you find yourself working alone mostly, or in conjunction with other developers?

AV: -- for the record, Dave Townsend recently fixed that bug.

AV: mostly alone

AV: I had to, actually. I may have learned computers on my own, so the idea of working with others was fairly new... and even now, sometimes that doesn't always work.

KA: When you have needed to communicate with others how did you do this? You mentioned you were on IRC, and you were using bugzilla-were there other methods?

AV: There's a number of different ways

AV: e-mail being the one we always overlook - in fact, Bugzilla wouldn't work without e-mail

AV: there's also written articles, sometimes editorials... and I think "blogging" evolved from the written-article/editorial concept.

AV: that's useful too, I've found

AV: newsgroups... one mustn't be ignorant of those

AV: technical documentation - thank goodness for people like Steve Rudman, Deb Richardson and Eric Shepherd, who have each taken turns running documentation efforts for mozilla.org

KA: Have you found any form of communication to be more useful during the development process?

AV: than any other? Not really.

AV: Each has their own advantages and disadvantages

AV: As Scotty said in Star Trek V, "How many times do I have to tell you, the right tool for the right job!"

KA: lol-right

KA: What about comments in the code-how important are they to the smooth development of Mozilla software?

AV: that applies in software development, too, by the way

KA: Yes

AV: personally, I appreciate the face-to-face meetings like the recent Dev Day in Mountain View... they're so rare, and they really give us a chance to expound and interact at a much faster pace.

AV: because let's face it... I'm a fairly fast typist, but the spoken word is much faster than the written one :)

KA: absolutely

KA: !

AV: seriously, I've had a very limited number of opportunities where I can just walk over a few feet and actually talk with the people who my work affects

AV: or who will be reviewing my patches, or documentation, etc.

AV: -- or, conversely, whose work I depend on.

KA: Right-something very different for the volunteer as opposed to the employee

AV: please, do not forget this point: everything I do ultimately depends on what dozens or perhaps hundreds of others have done in addition to me for mozilla.org, and for code projects like GNU C++ before and alongside them.

KA: Yes

AV: so while I'm tooting my own horn, understand that I couldn't have gotten here without some very smart people laying foundations before me.

KA: It's amazing when you think about the web of contributors it takes

KA: One of the questions we have been asking is for people would you describe your programming style? I notice this was something you ran into in your early experience working on DOM Inspector. Talk about that a little if you could.

AV: programming style... well, it's certainly evolved a little since my DOM-I days (I no longer actively hack on that), but it's remained fairly consistent

AV: the first step is really, "What's your objective?"

AV: What do you want to see working?

KA: Right

AV: half a moment; I'm thinking about how to phrase the next part, because there's a few different ways I go

KA: Ok

AV: brb

KA: Ok

AV: Actually, now that I had a moment to think about it, I can state that my programming style has always closely conformed to the famous "Scientific Method"

AV: namely: question, hypothesize, create test, experiment (write code, refine the code so it works), and write about it.

AV: it's always been that way, and the Scientific Method was my initial model for writing code. I simply forgot about the formal definition. :-)

KA: And have you ever found that this clashes with other developers?

AV: not the style or process, no

KA: Ok

AV: the clash comes at a separate step, which the Scientific Method doesn't cover: when others look upon your work.

AV: You remember my mentioning frustration?

AV: It's because I made so many mistakes early on.

KA: Yes

AV: I eventually coined a saying to summarize how I approach quality assurance these days:

AV: "The first step in confirming there is a bug in someone else's work is confirming there are none in your own."

KA: That seems like sound advice!

AV: I really can't overemphasize that.

AV: and even after you've filed the bug, never be unwilling to question the validity of your own test cases.

KA: If we could talk some about the volunteer process...

KA: To what extent has Mozilla relied on the work of volunteers?

AV: Mozilla would not exist in any form whatsoever without volunteers. Period.

AV: I know that sounds like a sound bite, like some prepared statement, but it's the honest truth.

KA: No, it absolutely seems that way

KA: Why did/do you volunteer and why do you think other people volunteer?

AV: Well, why do others volunteer? To scratch their own itches.

AV: Why do I volunteer? To scratch my own itch, which annoys the heck out of me :)

AV: I should note something here: I've never really considered myself a core member of the community - someone on the innermost circle.

KA: Ok

AV: most of the time, I feel like someone on the outside looking in - this despite the fact that I've now been involved with mozilla.org for about seven years now, and my work on Mozilla-based projects has directly led to the last two jobs I've held.

AV: err, the current one I have now and the last one I had.

KA: Why do you think that is?

AV: I'm not exactly sure - it could relate to the way I began as a programmer, as the lone wolf.

AV: it could also be that I have some rather different opinions about what's important :)

KA: In what sense?

AV: well, take Firefox for example

AV: Firefox is a great product

AV: hang on, I've got two different opinions here that want to express themselves :)

KA: Ok

AV: the beauty of Firefox is that it does a very simple thing, very well: it basically reads the web.

AV: it reads the World Wide Web, and presents the WWW to its users in a way that is simple, obvious, intuitive...

AV: Firefox has essentially raised the stakes for all developers. It's shown that power and user-friendliness don't always have to be in conflict.

AV: But...

AV: what were the three R's we were taught in elementary school?

AV: Reading, writing, arithmetic.

KA: So you see something missing?

AV: I do, and it's one reason why I was initially opposed to the Firefox effort branching off for a while

AV: ye olde Mozilla Application Suite (now maintained as Seamonkey Application Suite), had a Composer application built in.

AV: it may not have been the best, but it was there

AV: now, Daniel Glazman has undertaken to keep Composer development alive with his NVu project, and I applaud him greatly for that

KA: And some are working on Kompozer now, no?

AV: I can't speak much about Kompozer, because I don't know much about it

KA: But you feel in part Firefox is shortchanging Mozilla?

AV: au contraire

AV: I feel Firefox is turbocharging Mozilla

KA: lol

AV: but while Firefox development - and the toolkit underneath it - is proceeding at a fantastic pace, the ability to write the web in addition to reading it is falling far short

KA: Aaahhhh

AV: there are a number of different projects trying to solve pieces of it here and there, and I admit that one of those projects is my Verbosio project.

KA: I was just going to ask that

KA: If Verbosio was your response to this

AV: I mean, look around: MediaWiki, TinyMCE, NVu, AJAX authoring tools...

AV: it is my response, but there's a bit of history that I need to tell first :)

KA: Sounds great

AV: but to finish the thought I'm on, there's a whole bunch of people trying to write good tools for authoring content... only most of them are restricting themselves to HTML.

KA: Right

AV: Tactically, that's a good decision, because the 800 pound gorilla in the room, Internet Explorer, supports only HTML (and JavaScript, and more recently CSS) adequately... everything else web-content IE has just not picked up the pace on.

KA: Not unsurprisingly

AV: heck, I was saddened that IE 6 didn't support most of DOM Level 2, and even more so that IE 7 made no attempt to.

AV: anyway

AV: I started Verbosio because another authoring project of mine, Abacus, proved infeasible

AV: Abacus was an attempt to create a MathML extension for Composer. (MathML 2.0 is a W3C Recommendation for adding mathematical content to the WWW.)

AV: unfortunately, after I'd spoken at the Open Source Convention in Portland, OR about Abacus, I found out that truly extending Composer to support non-HTML markup was exceedingly difficult, if not impossible. To do it quickly would have meant incredibly ugly hacks, and to do it right would have taken years.

KA: Interesting

AV: not to mention that MathML is supposed to be in XML. It's fine if you're writing XHTML, but not ordinary HTML

KA: Right

AV: basically, it just wasn't going to happen.

AV: So I started Verbosio with the idea that we needed a truly extensible editor

KA: Wow

KA: That is a leap

AV: and unlike Composer, Verbosio would focus on XML.

KA: That would potentially change the way people thought about the web

AV: potentially -- I don't know yet if it's going to.

KA: If I could bring it back to volunteering for a minute-you said you do it to scratch your itches.

KA: So, is the work you do mostly for your own personal satisfaction or do you think about it in terms of the wide audience who use the products?

AV: it's kind of a mix... if it's not going to be useful to others, it won't satisfy me at all

KA: I see.

KA: Well, do you consider open source software projects as public service?

AV: I think you'll find that attitude among most software developers

AV: in a sense, that's exactly what they are

AV: public service... it means doing something to help others.

KA: You called Firefox a "turbocharged" Mozilla

KA: Why do you think Firefox has been able to attract a large number of users? What sets it apart from other open source projects?

AV: well, it had a different goal than the original Mozilla app suite.

AV: The original Mozilla app suite had a goal of being a standards-compliant cross-platform Internet suite. All well and good.

AV: Firefox had a goal of taking Mozilla technology and making it exceedingly friendly to users.

KA: Right

AV: both goals succeeded

KA: But to varying degrees...

AV: no, they both accomplished their goals completely. But the goal of being friendly to users meant users would like it more, would use it more, and would talk about it more.

KA: Ok-I see what you mean there.

AV: you'll have to ask Asa Dotzler more about that, especially about the "Spread Firefox" campaigns and the grassroots community that sprouted up from nowhere

AV: remember, I initially opposed Firefox on philosophical grounds. I saw it as a fork.

KA: Right

KA: Well, you have sketched these ideas out a bit in talking about VerboSio-what is your vision for Mozilla moving forward?

AV: for Mozilla, it's simple: to remain a solid platform for other people to develop products on, like Firefox, Thunderbird, Songbird, Cenzic Hailstorm, VerboSio, etc.

AV: for Mozilla Corporation, I'd be speculating, but I'd say it would be to support the community at large.

AV: I think XULRunner is a very important step in this process, and should not be underestimated

AV: -- or ignored!

KA: Have you contributed (as a volunteer or employee) to any non-Mozilla open source projects?

AV: Yes, but it would take me a few moments to think of which ones

KA: Sure.

AV: the company I work for, ManyOne Networks, is developing a competitor to Wikipedia which we call the Digital Universe.

KA: Interesting

KA: And it will be an OS model?

AV: my role in that has been to assist in developing authoring tools (do you see a recurring theme here?) and documentation

AV: DU will have several parts which are open-source, and potentially a few areas which are closed-source

AV: I can't speak with much clarity on the closed-source stuff, because it's still a bit up in the air

KA: Ok.

AV: also, I mentioned Website Abstraction earlier... although I haven't written articles for them lately, I moderate and actively contribute to the XML forum on its sister site, CodingForums.com.

AV: I also made an attempt to write a BigDecimal library in D (a new programming language intended to succeed C++ - <http://www.digitalmars.com/d>), because I've always been a big fan of mathematics and trying to do new things

AV: and a similar JavaScript-based BigDecimal library, which actually ran faster than the Java LiveConnect-bridged BigDecimal library for addition, subtraction and multiplication

KA: How did those experiences compare to your experiences at Mozilla?

AV: I'm not sure how to compare or contrast them, precisely, other than to say that with Mozilla, there's a lot more people I can interact with on a short time scale, so I can learn about how things work a bit faster

KA: Which I imagine can be satisfying

AV: basically, my talents as a software engineer have evolved as I continue to work on it, and Mozilla happens to be where my specialties lay now, because it's a platform I've worked with extensively. But I can still do other things.

AV: I should note that VerboSio, my XUL Widgets project, Abacus, etc. aren't really Mozilla projects; they're not sponsored or endorsed by mozilla.org. In fact, they're separate projects which I work on independently and MozDevGroup hosts.

KA: Ok

KA: Do you have experiences working on commercial products?

AV: I mentioned Cenzic Hailstorm before; it's a commercial product based on Mozilla code. <http://www.cenzic.com/>

AV: the Digital Universe may or may not qualify as a commercial product; that's something I can't talk about in great detail because I'm an engineer.

AV: and the marketing people would define that better.

KA: Ok

KA: How have these experiences differ from working on open source?

AV: well, for one thing there is proprietary code and concepts

AV: In the open-source world, particularly the free-as-in-freedom world GNU and the FSF advocate, such things are unfair

AV: but the way I see it, it's often necessary

AV: I mean, if software really was that free, no one would find any way to support a family on it.

KA: Probably true!

AV: Don't get me wrong, it's a laudable goal.

AV: but so was the ideal of Communism, or Socialism: that everyone would work for the betterment of his or her countrymen.

AV: sadly, that didn't turn out to be the case in the real world.

KA: True.

KA: How would you define a successful open source project?

AV: A successful one... that's a tough one :)

AV: Being successful involves two major steps

AV: one is achieving the goal the project set out to do

AV: the other is being widely useful among the general public

AV: the original Mozilla Application Suite achieved the first, but as I said earlier, did not achieve the second

AV: Firefox achieved both.

AV: will Verbosio be successful? I honestly don't know.

KA: What about projects like Apache that benefit the public, but are mostly unknown to them?

AV: touche :)

KA: Just out of curiosity

KA: They seem to fit your model

KA: But in a different sense

AV: I'd say that it's still useful, even if unknown. I know that sounds like a contradiction, but it isn't precisely.

KA: I agree

KA: Do you think open source techniques can be applied to other areas of production in today's society?

AV: I don't think it; I know it. Because if you think about it, open source "techniques" really came from ideals such as the Golden Rule (do unto others), generosity, public service... all concepts which we knew and understood centuries before computers ever existed. Certainly long before the Industrial Revolution and Mr. Babbage's calculating machine (whose name I can't remember now).

AV: we've had them all along; we simply applied them to the business of writing software.

KA: That's a very interesting take.

AV: or maybe I should say the "science" of writing software, since it really is a science, an art.

KA: Most of our responses have tended to flip it, and view the model as more of a free market idea, or look at the collaborative notions and focus on the physical difficulties of producing, say, an open source car

AV: all I'm saying it it predated the concept of a program.

KA: Yes. I haven't quite thought of it that way before. Very good idea.

AV: maybe I'm talking less about techniques than I am about ideology and community here.

KA: It is a vague question ;)

KA: So, given that-what's the future of open source?

AV: *shakes his head*

AV: now you're asking me to be a psychic :)

KA: lol

KA: A speculation if you have one...

AV: I think the best person to ask that question to would be Ray Kurzweil, who recently wrote a book I have yet to read titled "The Singularity Is Near"

AV: the future of open source, as I see it?

KA: Exactly

AV: barring some great catastrophe that plunges us into a new dark age, I'd say it's going to thrive, but we really don't have a clear idea what it's going to look like.

AV: I mentioned the Golden Rule a moment ago; people take pleasure in doing good works for others, and the same applies to software development.

AV: that said, the human capacity for imagination and invention is limited only by what we know now.

AV: it will be limited in the future only by what we will know in the future.

AV: the challenge for us, for all of us in software and for those outside it, is to figure out the ideas most useful to the general public and implement them.

AV: -- and if we happen to make a little money and support our families doing so, so much the better.

KA: Great-anything else you would like to add?

AV: I'm strongly tempted to talk about Verbosio more, about its goals for an 0.1 release, but I'm not sure that's relevant here. There was one other thing... let me reread the transcript to see if I can remember it.

KA: OK-feel free, we can discuss whatever you would like...

AV: here's something: how many years of college education do you think I've had?

AV: or what level of degree do you think I have?

KA: Well, I might guess none since you mentioned you were out of the Navy by 20

KA: Unless you have been back since...

AV: *snaps his fingers: you caught on to that, huh?*

KA: Historians are sticklers for details :)

AV: yeah, I've actually not spent a single day in college, unless you include the Journalism A-school that I spent six months at the Defense Information School (DINFOS).

KA: That is pretty impressive.

KA: considering all you have done

AV: that said, it has been a barrier to me in my professional career - these days, entry-level programmers need either a degree or a miracle to land a paying job.

KA: That's interesting-a change from the past I would imagine

AV: I had a miracle (a placement agency that contacted me). I strongly recommend anyone entering into the software job market don't rely on miracles.

AV: even now, I want to go to college and get that degree, if for no other reason than to say I did it.

KA: It's always an option....

AV: true. I just need to find a way to do it and not end up owing a few thousand dollars afterward :-)

KA: That is the key!

KA: What about Verbosio-what are your hopes for 0.1?

AV: For Verbosio, the goal is a simple one: Verbosio should be able to edit Verbosio's source code. In software development jargon, this is called "eating your own dogfood" - using your product on a regular basis to work on your product.

AV: that's the 0.1 goal.

KA: Is that in addition to also functioning as an html/xhtml editor?

AV: not precisely, and this takes a moment to explain

AV: Verbosio isn't intended as a XHTML editor, or a XUL editor, or any other type of XML language editor. Instead, Verbosio aims to be an extensible XML editor, where individual XML languages have Verbosio extensions written for them.

AV: now, people have made noise for years about wanting a XUL integrated development environment (IDE). I've wanted it myself. Verbosio will not meet that

goal, but for the purposes of a 0.1 release (not ready for end-users!), the two goals happen to coincide beautifully.

AV: Verbosio won't be that XUL IDE, but someone should be able to build a XUL IDE using Verbosio as a platform... much like Verbosio itself depends on XULRunner and the Mozilla platform.

AV: does that make sense?

KA: Sort of, though it is quite a bit over my head to be honest.

AV: okay, for Verbosio 0.1 to meet its dogfood goal, it has to be able to edit XUL.

AV: effectively.

KA: Ok

AV: it also has to have an understanding of XBL, some JavaScript, some CSS... a bunch of other little things, and especially about what a XUL application is.

AV: these requirements also happen to be necessary for an IDE that focuses on XUL.

AV: so it's a nice parallel for the 0.1 release.

KA: Aaahhh-ok I think I get it now

KA: So this will be extremely useful for developers...

AV: now, currently Komodo IDE and Komodo Edit from ActiveState are the best free tools I know of. Eclipse is probably better, but I'm not familiar with it and it's more oriented for Java development.

AV: I think it will be. My whole track of software development has been along the lines of building better tools.

AV: Verbosio 0.1 will not be better than Komodo IDE or Komodo Edit.

AV: I know that going in.

KA: Right

AV: but it will offer some capabilities that Komodo doesn't currently have.

AV: post-0.1, I'm going to stick my head up for air, and see what other people are doing in this field.

KA: And it hopefully will tap into one of the largest development communities out there.

AV: I may decide to merge with other projects, including quite possibly Komodo (even though it is closed-source), or I may work on a 0.2 release which focuses on reliability and quality assurance (making sure that Verbosio handles XML correctly).

AV: or I may find that there's already a far superior tool out there - though for XUL development, I rather doubt that. :-)

KA: Right.

KA: Well, thanks so much for taking the time to chat with us!

AV: thanks for having me - as I said, I don't consider myself a core community member, so I am flattered someone's willing to chat with me for posterity.

AV: -- for three hours!