“Usability” of the Perl Online World for Newcomers

Shlomi Fish <shlomif@shlomifish.org>

Revision History

Revision 540	2003-04-18	shlomif
Removed a paragraph that is less than accurate, added a link to ESR’s CatB, and refactored a few things.		
Revision 536	2003-04-16	shlomif
Made the criticism against the lack of availability of the O’Reilly core Perl documentation less blunt, and added a suggestion for the www.perl.com site. Fixed some typos.		
Revision 535	2003-04-16	shlomif
First Advogato Version.		

Table of Contents

Me as a Perl Newbie

“Shlomi, it takes 10 years to learn UNIX. I want you to learn it in a month”. My future supervisor in a web publishing house I switched jobs to, told me that and It’ll be a cold day in hell before I forget this quote. He also showed me Perl, telling me I can write a TCP/IP client in 5 lines and a TCP server in 10, which sounded very impressive and intriguing. So I set out to learn UNIX and Perl.

When I got to work I did not know UNIX as much as he could hope for, but with some help from the people around there I managed to get by. The year was 1996, it was the start of the Internet Boom and everybody and his mother wanted a web-site. Perl 5 was there, and it was practically the only thing you could use on UNIX to write CGI scripts, without completely losing your mind.

I only had a Windows 3.11 at home, and so started learning Perl from all the POD documents cramped into one Acrobat Reader file. "On dry", as they say in Israel, without actually running it. I understood some of it as it reminded me of C and BASIC which I knew quite well. But still there were all these references to sed, awk, UNIX shells, and other things I did not know. Generally, they assumed you knew what they were talking about.

I started working there, and was able to get by with my little knowledge of Perl and UNIX which accumulated quite quickly. I remember asking my supervisor what regular expressions where all about and he told me: "they are a mechanism to find patterns in text". That was the enlightenment that made me able to use them on a regular basis.

I fell in love with both UNIX and Perl. I was not much of a UNIX guru and wrote what I now write in shell in Perl, and simply kept asking my supervisor (who was then a true UNIX guru even in today's standards), questions about how to do things. But I got by.

Since then, with some breaks, I extended my knowledge of both Perl and UNIX. (I was thrilled to find ActivePerl and use Perl effectively on Win32 as well). Gradually, I learned about Modules and Objects in Perl and then a lot of tips and shortcuts, and now use it very often. Even in the Technion, where I now study Electrical Engineering, Perl proved to be a life saver. It is my favourite language by far, and would not replace it with anything else.

That's my story. Now let's move to the story of Mel Jones (short for Melissa but might as well be Melvin), a Perl newbie who wishes to get hold of Perl. When thinking about the usability of computer systems, it is helpful to think about a few imaginary figures, which one is almost certain there are many similar that exist in real-life. Mel is one of them, but as you'll see later she won't be the only one.

Before I start let me say that I am indebted to Larry Wall, Tom Christiansen and all the other people who contributed to Perl and brought Perl so far. I did not come to bury Perl or the Perl world in this article. I simply came to pass some constructive criticism and suggestions to further aid its public acceptance. Perl is the nicest language I've seen so far, and I would highly recommend anyone to learn it.

(I am also grateful to the Python, Ruby, Tcl, PHP, etc. crew for working on languages which can provide a viable alternative to Perl in some cases and by some people. By all means, Perl is not suitable for anyone and for any purpose and I always appreciate a healthy competition.)

The Story of Mel

Who is Mel? She may be an experienced C programmer, or has some working knowledge with Visual Basic, JavaScript or ASP. In any case, she does not know Python, Ruby, Lisp, Tcl or PHP. (such people will usually have little problem learning Perl). Mel got a new job: be it as a system administrator, CGI programmer, Bio-informatics engineer, hardware designer, QA engineer or whatever. She wishes to do something that seems awfully hard with what she already knows, asks around and receive a simple answer "Perl would be perfect for it. And it is spelled P.E.R.L"

So Mel goes on looking for this strange and perfect language called "perl" (without the 'a'). Where does she go first? Google [http://www.google.com/]? Most probably. What does she discover? www.perl.com is the first hit. An overcrowded site with a lot of O'Reillyisms with several articles that are way over her head. Perl Mongers is the second hit. She reads on - groups of Perl users - goodie. Now if she knows how to use E-mail (she may or may not have such a good net-wisdom), she may have hit the jackpot, because

most Perl monger groups are very friendly to newbies. Then she discovers CPAN (Mel says: "What is it and why do I need it?"). ActiveState (a commercial site which may or may not scare her) and Use Perl, which is a nice site for Perl News, but she is naturally interested in the Perl's status quo.

She refines her query: "learn perl", "begin perl", "perl newbie" or "perl beginners". The concentrated site for perl beginners is learn.perl.org [http://learn.perl.org/]. It contains some reviews of books (which Mel may not have time to read), a free online book (again, same issue) and no links to tutorials whatsoever. IMHO, the site's visual side is also very lacking and she may come to believe it is a sub-standard site as a result.

OK. Maybe Mel was given the Llama book on the first day of her job as a gesture from her employer. Maybe this employer has these books online on his intranet for everybody to see. (there are also illegal or unfirewalled mirrors of these books, but let's suppose Mel is a lawful citizen who respects copyrights.) The question still stands: why do you need a book to learn Perl? Why can't you learn it from the Internet alone? Why kill trees, when you can simply transfer electrons from place to place to study it?

The Situation Today

When Larry Wall and others started hacking on Perl 5 (which was a complete rewrite and complete re-design of the defunct perl4), they expected it to be used for system administration tasks, for text processing and maybe to automate a few network tasks. The Internet Boom wanted otherwise, and several other technological explosions saw in Perl a suitable language for its tasks. Nowadays, Perl (and similar languages such as Python, PHP, Ruby or Tcl) are used for tasks as diverse as the computer world itself. Server-side- scripting, Bio-informatics, System Administration, Text Processing, System Tools, Graphical User-Interface, Computational Programming, Quality Assurance, automating Internet services (for good or worse) and even real-world applications and Games, are all written in Perl, Python and everything else. Some Killer apps of this kind include: Slash (the back-end of Slashdot), SourceForge/GForge, Bugzilla, Zope, PySol, The Mandrake Linux System Tools and Installer, Frozen Bubble [http://www.frozen-bubble.org/] (;-)), SpamAssassin, and many other things I don't recall or forgot about momentarily. Furthermore, a lot of code of such languages is used internally and not exposed to the world, and plays a large role behind the scenes.

They are still often referred to as "scripting languages", but this term can no longer accurately describe them. While a Java-obsessed friend of mine asked me to give him an example for a proprietary application of a large codebase that is written in Perl, and I could only think of open-source ones, it is definitely enterprise ready, and not a toy language for hackers any more.

Now, let's go back to Mel. She needs to learn Perl. But she might as well have been told to learn PHP, Python, Visual Basic or whatever. My problem is I think she would have a much harder time with Perl than with the others. No, it's not because Perl is a harder *language* than the others. I don't think it is. It's because the resources available online are inadequate or may scare her. And as Casey West noted in a previous attempt at correcting the situation [http://www.perl.com/pub/a/2001/05/29/tides.html], the Perl community is relatively hostile to newbies and does not put up with them as much as those that are friendly would hope.

Here is what I think are the problems with it:

Concentration of Effort and Elitism

I did my best to help newbies welcome in Perl. I actually like beginners, and have a lot of patience to them, because I know no one is born with an innate knowledge of Perl, UNIX or whatever. I often work in the Technion farms and whenever I overhear that someone has a problem my ears become pointed, and I cry "do you need help?" And then show him how to do something that for me may be perfectly obvious. Seriously. I simply enjoy doing that.

I also took some proactive effort in the direction. I wrote the Perl for Perl Newbies [http://vipe.technion.ac.il/~shlomif/lecture/Perl/Newbies/] lecture series for the Haifa Linux Club. I wrote them with absolute beginners in mind, and decided to teach a very small subset of Perl step by step, with assumption that people will read the man pages afterwards. (and be able to understand them better then).

Lately, out of frustration with the learn.perl.org site, I started working on my own version of a Perl beginners site [http://perl-begin.org/]. Now here is where things went a bit wrong in my opinion. However, I don't want to defame anyone so I'll summarize. Both the use.perl.org [http://use.perl.org/]'s editor and a very prominent Perl figure that stumbled upon the site, believed that I should merge my work with learn.perl.org. At first, I thought it was a good idea, but then discover I am unable to contact the learn.perl.org workers and that their mailing lists was closed for subscription.

Regardless, what's wrong with two Perl beginners site on the Internet? If I feel that learn.perl.org is inadequate, why should my effort be frowned upon? If we take PHP or Visual Basic for example, than everyone and his mother have a PHP portal where users can post questions (sometimes without subscribing), receive answers, read tutorials and see code snippets. Same for Visual Basic, despite the fact that it is proprietary. I want the same for Perl! I don't want a couple of concentrated perl.org and perl.com and oreillynet.com sites. I want a real network of independent sites!

It also demonstrates the Perl community's elitism. I don't know how it came into being. For example, when I log into EF-Net, the channel #perl displays a long title that reads something like: "Not a Helpdesk! No CGI/WWW/Net::IRC/FAQ.". OK, there's also #perlhelp where such questions are accepted. But here are two facts from a user interface designer point of view:

1. People are not going to read the title. In a user interface, you are lucky if people read anything at all. (including: "Delete and Expunge 'My most important work'?")

2. Newbies by nature will try #perl if they log into a random IRC server. And they don't want to get flamed for being off-topic there.

In short, this is a bad UI design. If the Perl gurus so desire not to be disturbed by newbie questions, they should move to #perlgurus or #perlelite or #perlcafe or whatever. But they should not make claim for #perl.

In another IRC network, I (not a newbie by far) was kicked because I pasted a URL and said the word CGI. (there's a nasty pattern bot there). Some other time, I was kicked out of #perlhelp, because I said the built-in function hex() converts numbers from hexadecimal to decimal, while a moderator believed it converts them *to* hexadecimal. Naturally I was right (Run perldoc -f hex), but still instead of arguing with me, he kicked me out. Why should a bit of disinformation hurt if there are enough good people to correct it on the IRC as well? And Newbies can learn a lot from such discussions.

Lack of Adequate Online Resources

Let's face it: at the moment, there aren't too many good Perl tutorials online. Many people who ask trivial questions are being conveyed something like "Read the Llama Book, the Camel Book, the Black Panther book, the Perl Cookbook, and then you'll know how to convert a file to lowercase." Maybe I'm exaggerating a little, but it seems people are so content with the books that they don't seem to think people who don't want them, cannot afford to buy them, or don't have time for them, are worthy of learning Perl.

I don't have anything against people trying to make money off Perl by selling books. But these people are the same people who are the Perl project leaders, and so are responsible to make sure Perl is well-documented. If people get frustrated learning Perl, and become unhappy with it, they will criticize "Perl". Not the Perl documentation. Not the Perl community. Not even the Perl leaders. "*Perl*". If Larry Wall et al. care enough about Perl, they should make sure it has good online documentation. They don't have to do it themselves, as a simple call for action by a very prominent Perl figure will do. (in accordance to Eric Raymond's "The Cathedral and the Bazaar" [http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/])

If we take a look at Python or PHP for instance, then they have excellent online documentation, which is very useful for most if not all newbies. No need to read a book or take a course. Just surf the Internet on your ADSL connection, install a few things on your Windows machine, and presto! You know PHP or Python or Ruby. I seriously doubt if it can be done with Perl with the same ease.

I am grateful for Simon Cozens for making his *Beginning Perl* book online for free. I did not take a thorough look at the book, but it is more encompassing than anything I've seen. But even one book is not enough. People still want step by step tutorials and things they can cut and paste code out of and tinker it to do something.

When are books that cover a Technology something that I find legitimate? When they are not absolutely needed. I learned the Perl DBI by reading the man page and the online site and asking a few questions. Some people do not like man pages too much and would rather buy a book that covers Perl DBI. That's OK, and there may be a few fine points of working with DBI that I missed with what I call my "bottom-up learning".

Such books need not be available online if their authors so much don't desire. However, Perl is very hard to learn from public electronic resources alone. I believe there may even be a clash of interests because the core Perl people also write them and so may not have enough motivation to improve the online documentation. Making them public will resolve that.

That aside, I also believe that making it available online will not dent their sales much. I still would like to buy the Camel Book (which I borrowed from the local Perl Mongers library), and loved regardless if it's online so I can read it at bed time and stuff. And I generally have no problems using a computer book or tutorial.

Lack of Adequate Mailing Lists

Then there are the mailing lists. I think (but am not sure) that most Perl Mongers mailing lists will gladly accept questions from newbies and answer them kindly. I know this is the case for Israel.PM [http://www.perl.org.il/]. As a general rule Perl Mongers should make sure that the newbies mailing list would be the first list they see on the site, or try to subscribe to. Again, good user-interface design.

The Perl Beginners list In my opinion is a bit useless. It is very high volume, and usually whenever one considers replying to a question, it receives three replies by the time he finished composing it. I was also subscribed and it was too high volume to be effective for me as a "guru". I believe newbies will not be able to handle such volumes either. Unless they can configure their mail readers to filter only their own threads, which I'm not sure they can. Perl can do that, but they need to know it first.

Like I said earlier, I think distributism is the key to our success here. Just like there are many scientific journals with very similar names and themes, (sometimes, "Physics A", "Physics B", "Physics C" etc), so there should be several international Perl beginners sites, each with his own mailing list for beginners. Gurus will subscribe to one or more sites and answer questions there, and newbies won't have to excessively use their delete button. Gradually, each such community will have a different atmosphere, and we will get a more diversified Perl world. (we all agree that diversity is good, right? If you don't you should program in Python)

Substandard Standard Documentation

Let's face it, the man pages (or perl*.pod), which are the definitive resource on the exact behaviour of Perl are very bad. Maybe they were OK for expert Perl 4 or Tcl hackers, or those that knew UNIX shells, sed and awk well. But nowadays we can expect people to come even without knowledge of the DOS or Windows command line, much less UNIX wizardry.

Perl is not and cannot be a "point-and-click-no-think" language. I'm not going to fall into the trap of trying to make it so. Even the lamest GUI kiddie eventually learns to invoke a text editor and write some glue code

for whatever he needs to get a web-site or whatever working. Perl is much too powerful, to be something like ASP, where you can write wizards to perform most common tasks. In Perl, you can in one script, write a full application. This is something that while perhaps possible in ASP, it was not intended to do.

Still, the fear of writing code and thinking in terms of variables, functions, objects, modules and other abstractions is not something that for people above a certain age is easy to overcome. Visual Basic for Applications is one of the things I like most in Excel, but most people don't know it, and most books don't teach it. Why? Because it involves actual coding and people don't like that. Ironically, it's very useful, too.

Now, a documentation should guide a newbie step by step and explains: what is a variable? what is a scalar? what is a number? what is a string? how they convert into each other? What is an array and what it is good for? Same for hash, object, etc. All these things may have different semantics in Perl than in most other languages. And all these things need to be explained clearly and reasonably, without depending on an analogy to tools that are way over Mel's head at her current position.

I don't want this knowledge to be hidden from the world in some proprietary O'Reilly books. I did my best to remedy it, and Simon Cozens did even more. I think the reason people think Perl is such a poor language to learn or even for complete beginners, is not because of the syntax and semantics themselves, but because of its poor documentation. That put aside, I think publishers making their books available online, will only help them sell more books (assuming these books are good, of course). Otherwise it's similar to buying an apartment without seeing it first.

Too Much Hubris

Hubris is one of the three great virtues of the programmer according to Larry Wall. But he also noted in "Open Sources" [http://www.oreilly.com/catalog/opensources/book/larry.html] that humility was a good quality as well. Don't get me wrong some projects have much more hubris than Perl: Python and Linux come to mind. I'm not saying all Pythoneers or Linuxers are hubris-ful. But a small minority is too loud, and there's a similar case in Perl to a lesser extent.

What do I mean by hubris? "Perl is the greatest language since sliced bread. It has unparalleled text processing and data structures support. It is object-oriented, supports functional programming and any other programming paradigm very well. It's perfect for almost any task, from powering web-sites to writing GUI to writing games. It's practically LISP with in-fix notation, and runs on almost any platform imaginable..."

You get the drift? Why I think it's bad? Because most people don't like communities with this sales pitch. When recommending Perl to someone, you should discuss one advantage at a time, not overload him with a lot of information. When people see a lot of information they think there's something fishy about this.

Let's take a look at a language with just the right amount of hubris: Visual Basic. Right, a language that Microsoft has released several versions of, each adding major things, but is still inferior to perl 5.000. A language whose code can only be deployed on Windows. A language that is many times useless without a whole slew of OCX or OLE components that are freely available at best. Still, Microsoft and the other VB community do not claim it's the greatest language on earth. "I take your word that Perl/Python/Ruby/LISP/whatever is much superior. VB is stupid, but cute and it gets the job done." That's the attitude we should have: Perl gets the job done. Granted, many different jobs. But as far as Mel is concerned it only gets her job done and not the entire world's ones.

What's the threat?

You can often hear opinions or wonderings in the vein of: will PHP kill Perl? Will Python kill Perl? Will Java kill Perl? Will FooLang kill Perl? I don't think anyone of them can hope to. I don't think Perl will completely eliminate any of them either, but I think Perl's status is secured.

Perl is too syntactically and semantically rich a language to be abandoned by all its loving users in such a haste. CPAN is also a big plus for Perl, which while being a purely technical mechanism makes Perl more suitable for some tasks than most of its other competitors. I hope to see equivalents in other languages, but it will take some time and a lot of work from some very determined individuals to get it up.

The Perl Culture has extended way beyond its use. From perl poetry to Perl golf, to the talks by Larry Wall and other of the very interesting Perl personalities. The Perl community is alive and kicking and proud of being such. While some people abandon it for Python, Ruby or whatever, most of them still think Perl is a nice language, and don't abandon it or completely or completely antagonize it. The beauty of the Perl culture is that it's about using the right tool for the job whether in Perl or not.

We all know 80% of the users use 20% of the features of a given program. However, as Joel Spolsky identified [http://www.joelonsoftware.com/articles/fog0000000020.html] they often use a different set of features. Perl's feature-set is huge, but you can survive with knowing maybe 10% or even less of it. Moreover, you can gradually learn more and more features and keep learning new ones. Perhaps only the three authors of Programming Perl know the entire feature set, and I believe even they are actively unaware of some obscure details of it. However, one always feels he's getting closer to 100%.

The problem is that Mel is not concerned with any of it. She just wants to write a program. This program may just as effectively be written with Python or Ruby or PHP. And we should be concerned with her writing it in Perl, because otherwise, it will take her more time to get to learn Perl. As much as I care about the quality of Perl programmers (which is actually quite good) I also think we should evangelize it to many people who'll find it useful.

Moreover, I don't like Mel becoming one of those characters who say: "Perl? This horrible language? I'm using PHP/Python/Ruby/Java. It's much better than Perl." I don't think there's a way to exactly measure which language is "better". Some languages are better in some regards and for some purposes than others. But Perl is much more than a language. It's a culture and a philosophy.

Perl has a lot of potential for Mel. Alternatively, it has such for Joanna Simpson, a CEO of a Fortune 500 firm who is trying to ship her automobile-company's new car model. I think we should even try to teach Joanna Perl, because not only will she learn how great a language it is, but it may actually be useful for her in her day for day work. Otherwise, she would tell Robert, her secretary, to sort all the files into several directories a week from now, which can otherwise be done with a 10-liner Perl script. This will leave Robert enough time to answer phone calls and play Frozen Bubble [http://www.frozen-bubble.org/] (which is written in Perl) in between: things that Perl or any other software cannot do for him.

Perl should be everywhere. Let's not ruin the day for Mel, Joanna or Robert by a bad attitude. Remember that at a time, we were not much more educated than them. Not even old time UNIX gurus like Ken Thompson or Larry Wall can claim that, because UNIX and Perl developed based on input by many people and everyone was always surprised by new developments. While the effort was concentrated, the ideas flowed from everywhere.

Furthermore helping newbies is *fun* and makes you feel good. You don't have a moral obligation to do so, but I believe you'll find helping someone even with a trivial question even more enjoyable then getting help for something that bothered you. That's how the open-source world works.

You may eventually find yourself in a similar situation in something else entirely, and may need help by people who are more expert than you are. During my Electrical Engineering studies I studied things that were much less straightforward to me than either Perl or UNIX (which I don't consider easy either), and sometimes the help of the lecturers, the T.A.'s or fellow students proved crucial to understanding them. Moreover, explaining something to someone who grasps it much less than you, is the best way of learning (even better than actually experiencing with something yourself). That's because when teaching someone, you have to think how to convey it using more basic knowledge in a structured way, and so understand it much better.

What we should do about it?

Until now it has been a long rant with some sprinkled advice in between. Now let’s get to the real action. Here is what I think should be done, in no particular order.

Independent Activity

Establish your own Perl beginners site/portal/mailing list/web forum/etc. learn.perl.org is not enough and neither is what I’m going to do. We need more and as many as possible.

Note that if you feel that there isn’t enough traffic or that you should merge with some other site - it’s OK and you can do so. As always reason and experience are your best guidelines as there isn’t a precise rulebook for surviving in real life.

Become your biggest Perl Guru

Remember that the biggest Perl guru is *you*. Not Larry Wall or Tom Christiansen or Damian Conway or whoever. That’s because none of them can dictate to you how to work with Perl, solve problems with it or with something else, etc. As much as I appreciate their efforts at making Perl better and more usable, I don’t think they have or should have a monopoly on where Perl is going to. The world is your oyster.

Treat Beginners Nicely and Help Them

Be kind to newbies, answer them accurately, don’t confuse them and try to use Perl that is as simple and non-idiomatic as possible (e.g: for the file’s lowercasing example, don’t use Tie::File, and prefer lc() over tr///).

If you don’t have the time to write the entire code for the beginner - politely tell him you can’t help him at this point or whenever. Else, try to actually write some code which would be as clear and commented as possible. Don’t expect beginners to like Perl Golf snippets. They may mistake them for line noise. ;-)

Think about Usability Issues

Think about the User-Interface Design of the System. What will the user assume? What would be the first thing he’ll do? What is he going to notice? What is he going to ignore? What is he going to do mechanically? These are all questions that are as pertinent to IRC networks, programming languages or web-sites as they are to Microsoft Word.

I gave the IRC channels examples of bad UI design. OK, gurus, once upon a time there weren’t too many newbies around. But if you don’t want to completely reject them, it is a better idea to move to a different channel.

I believe the Perl Monks site also suffers from such a lacking design. First of all - its name. Do you honestly expect someone to be a Perl “monk” and live in a Perl “monastery”? (I’m Jewish, an Atheist, and very sexually liberated for God’s sake!) Secondly, the site’s design is unattractive, overcrowded and confusing. I remember I was reluctant from participating there, and had troubles understanding what it was all about.

Web forums should involve no registration whatsoever (just fill in the name and E-mail), and not use too fancy a markup syntax, at least not by default. I actually saw a forum (not particularly Perl-related) where markups were placed in brackets, and I wanted to paste some Haskell code... I could not understand how to do it, no matter what I tried .

Create Archives of Perl Code and Documentation

Publish some of your Perl code online. Establish a Wiki. Don't ask anybody for permission - just do it. There will be chaos first, but order will emerge out of it eventually. And if not, Google and friends will like us better. ;-)

Rik van Riel of the Linux Kernel fame, told me and some IRC folks about a system which was written for internal use by Connectiva (his former workplace), that uses a DMoz [http://dmoz.org/]-like category tree, where Slashdot-like comments are added. This sound like a nice integration of a few ideas. The Connectiva code is written in PHP and is not ready for release. Nevertheless, I think tweaking Slash [http://www.slashcode.com/] to do just that should not be too hard, and may prove as a useful alternative or a complement to Wikis.

Don't be afraid to charge money or say no

Let's suppose that despite all the positive changes and having a book nearby, Mel feels she needs a face to face guidance. She asks around her workplace, and discovers a fellow worker, by the name of Rachel Southern (actually a character in a story I'm writing who fits perfectly here) is a first-class perl guru. Rachel has been programming since she was 14, knows UNIX, Perl and many other languages well, and is a very nice, sociable person. So Mel asks Rachel if she can tutor her in Perl.

Rachel, however, being the expert that she is, is busy with work, and prefers to spend her weekends relaxing, socializing (online and in real life), and hacking on open-source projects. Moreover, she lives very far from Mel's apartment. So she tells Mel that if she wishes to go to Mel's apartment on a Saturday or a Sunday, and tutor her with Perl (or UNIX in general), Mel would have to pay her a substantial amount of money per hour. Mel accepts, as she realizes the help she'll get out of Rachel, will help her in her work, which in turn will be very well worth the money.

Was Rachel "greedy" or "unfriendly" in asking for money? No. I was told a technician in Israel charges $50 an hour. Many lawyers and other professionals will charge much more. The time of competent persons is the most precious resource on our planet (except perhaps for human rationality), and Rachel has no selfish interest to spend so much time helping Melissa.

I was criticizing the fact that books costed money and were not available online only because the reproduction of books in Electronic form is zero. But a time a person spends helping another one is priceless. Now, if Mel lived in the same block as Rachel, or otherwise they were very good friends, then Rachel could have chosen to reject the payment. (My friend and I lost track of how even we are regarding the pizzas we ate and we could not care less about this fact).

It is true that Richard Stallman (RMS) travels around the world visiting various free software user groups, at the cost of the travelling expenses alone. He have also consistently answered the E-mails of me and everyone else I know, and is available by phone most of the time. However, Stallman's purpose in life is propagating free software and he is the spiritual father of the free software and open source movement. As such, he accepts the fact that he has become a saint and does not charge money for that. As he once told me and some other IGLU [http://www.iglu.org.il/]'ers: "Advocacy is everything I do".

Most of us, however, are trying to make the best use of our time, including getting the bills paid or contributing to free software. As such, our time is precious enough so a substantial amount of it that is used to help a less capable peer will cost money. It does not mean that spending some time for writing a straightforward ten-liner for someone should, though. (in that case, there's usually no way to acquire the money in the first place)

There's a limit to anything. I remember that once I hanged on the IRC, and someone there showed me a bot he or his clique wrote in mIRC-script and wanted to convert to Perl. The bot was a bit complex, so I told him I did not want to convert it, as I had no selfish interest in getting it ported. I suggested him to

learn Perl and do the porting himself. He understood. I'm not sure the bot was ever ported, but the gods help only those that help themselves.

Revamp of the existing resources

Like I noted earlier, I believe many of the existing resources, while being perfectly fine for veterans may scare away newbies. www.perl.com is crowded with too many links, too much content, and too much oreillynetism. Since it is the first hit on Google, it should be a decent page where newbies can easily find their way in. Something like the Python Homepage [http://www.python.org/] or PHP's [http://www.php.net/]. Plain, simple and effective. (note that I don't have a problem with the O'Reilly Net format as a general rule, it's just not suitable for newbies who may have never heard of O'Reilly or do not have the time to put up with it.) The Python.org site, for instance, contains just the right links for a newbies and nothing more. It has a rather cheesy site, but it is much more usable than www.perl.com. The latter has links to a host of other O'Reilly sites (what for), some articles that are way over Melissa's head, no explanation of what Perl is all about, links to many O'Reilly books (again: why?), and is generally overcrowded with information. This is not a community site. It is practically a blunt promotion of everything O'Reilly!

www.perl.com as it is can be renamed as perl.oreilly.com or whatever. But otherwise, O'Reilly should not make a claim for the homesite of the Perl language and Perl community as they are not the only player in the game.

Also, I kindly ask the authors and publishers of the core Perl books to place them online. If a book teaches you advanced things that you can learn on your own, that do not absolutely require a book (like Damian Conway's "Object Oriented Perl"), then putting it online would not be necessary, but still a nice gesture. However, it cannot be said on *Programming Perl* (the Camel Book), *Learning Perl* (the Llama Book), the *Perl Cookbook*, or *Advanced Perl*. I could not care less however, for *Perl for Oracle DBAs*, because it's highly specialized and of no interest to Melissa or Robert. Rachel can easily learn it on her own should the need arise, and Joanna's knowledge of Perl should not extend to this speciality.

There are other popular sites or resources, that may seem quite hostile to newbies. Perl Monks is one. use.perl.org should actually be quite good. Its standard Slashdot-like seemingly-underground look can actually appeal to newbies who wish to keep up to date with Perl news. What the editors should do is answer newbie submissions themselves should there be such, and *only then* tell them that it's the wrong place to ask. Remember, people don't read Submission Guidelines or whatever, and even I act mechanically on many web-sites. (don't you?)

Generally, if one of the new resources would be better than the existing ones, then it would eventually supersede it. Yahoo and AltaVista were superseded to most extents by the much superior and more user and community friendly Google and Mozilla Directory, and we should not fear a healthy competition either. We should not concentrate our efforts, because we will also reduce the number of people doing them in the first place. And as Eric Raymond noted in *The Cathedral and the Bazaar*, in a distributed Bazaar model, the output grows linearly to the number of people involved.

While the Perl's source code was maintained in a very much Bazaar-like way, the Perl Community has been relatively Cathedral like looking for a selected small number of shrines. Let's change that.

Conclusion

Perl is going strong and will continue to grow in the nearby future. However, many beginners are deterred from becoming part of the Perl world, or understanding it, out of several defects in the online Perl community and resources.

Perl proves to be a gateway to UNIX for many people. Many people who start to use a UNIX-compatible system (like Linux or Solaris) write their shell-scripts in Perl, and later on actually learn shell programming.

Many Windows people who used Perl for Win32, find the UNIX concept much more desirable afterwards. Perl is a reflection of UNIX in all so many ways, and I’d hate for someone to get scared of UNIX as a result of getting scared of Perl too.

You often hear people complaining at Perl’s briefness, TIMTOWDIness, difficulty to learn, “inconsistency”, “ugly syntax”, complexity, in-fix notation, size, dollar signs, lack of suitability for large codebases, etc. All these “issues” are very much marginal if not completely false in getting it into public acceptance. In fact it is a sign that it has a culture that not all people can accept, which is good, because not all people are the same in such amoral issues as choice of programming language.

However, what can deter someone from learning Perl is a lack of good support and aid from an online community, which may be the only connection he has to this virtual world called “Perl”. Let’s change it. Let’s make sure Melissa will be practically sucked into the Perl world, as any community happy to receive a shining, new, interesting member.