

Keyvan Nayyeri

God breathing through me

<http://nayyeri.net/>

Slides of My Presentation on Scala

Beginning my studies here in the US and for the first semester, I'm assigned a Teaching Assistant position for an undergraduate course on Programming Languages. One of the major goals of this course is to get students familiar with major features of different categories of programming languages as well as the differences in implementation of some basic features.

The instructor of the class, [Dr. von Ronne](#), has decided to use Scala as the main programming language for projects defined for this class, and students are supposed to use this language to build an incremental project and compare the features of Scala with other languages taught in the class.

To get students started with the language I went over the principles of Scala and functional programming in two sessions at the beginning of the semester (back in mid-January).

As some of you may know, [Scala](#) is a programming language that combines the Object-Oriented and functional programming features from the two major paradigms of programming languages (declarative and imperative), and was founded and released in 2003 by [Martin Odersky](#), a German scientist. The name of the Scala language comes from Scalability and it truly delivers a high level of scalability. The language is designed to be expandable and flexible.

Scala has proven to be the best attempt in combining Object-Oriented and functional styles in a single programming language. The good point about Scala is that it doesn't come as a completely new language and just tries to refine the Java platform or the .NET Framework. Scala is more known for its integration with Java and the .NET branch hasn't been very active (most likely because F# has become the mainstream functional programming language for the Microsoft community). Scala has a compiler and an interpreter for Java and has a great set of features enhancing the functional programming style.

The other interesting point about Scala is that it derives many of its features from existing languages and tries to get the best out of the previous experiences.

I put all these introductory information in a set of slides for a short presentation in to start the class. I thought that it may be helpful to published them here for those who may be interested to have a background in Scala. You can download the slides in the PDF format from [here](#).

If you're interested to learn the basics of Scala in a practical manner, [Simply Scala](#) is a great website to use with its tutorials, and if you're looking to learn it in a more advanced level, [Programming in Scala](#) is a rich book written by some authors including Martin Ordersky, himself.

<http://nayyeri.net/slides-of-my-presentation-on-scala>

I know that it's been quite a long while that I haven't written anything on this blog and it marks a record in my blogging history. Moving from a country to another and starting a new life specifically as a graduate student has its own difficulties that eat all your time.

However, the good news is that now I have some obligations to blog at least once a week until the end of this semester and it gives me a good chance to return to blogging covering new topics that may not be very related to the .NET world!

For this semester I've taken a graduate Software Engineering course with [Dr. Macneil Shonle](#) (recently, we've been finding many clones of [Scott Hanselman](#)!) and as a main part of our duties for this course, we're supposed to work on a software project as a team.

The software project for this course is something that should run on top of the iPhone/iPod Touch platform and we need to form groups of 4-5 people. We're supposed to collaborate on this project and deliver it in a timely manner (by the end of the semester) and follow an incremental approach to build a useful application. Besides, we're supposed to prepare documentation and information about our work individually and specifically blog about our experiences on a regular basis. Surely, this is good news for my blog readers.

Apparently, Dr. Shonle had some goals in mind by defining this task for us:

Force students to work as a team and learn about all the team-work concepts Help them learn about new platforms and technologies especially with a funny project for the iPhone platform Contribute something to the community by building new applications. Learning Software Engineering concepts and applying them in a real-world project in action.

Therefore, I'm going to publish some blog posts in the next three months on this blog and cross-post them to our internal Moodle blog to get some points! I try to share my experiences, opinions and the challenges that I face with, so my readers can grasp an understanding of the process of programming with the iPhone/iPod Touch platform.

To be honest, I'm not a big fan of Apple and haven't used their products that much, and I'm not planning to do that for ages. I know that there are many people who find their products exciting but as a matter of a fact, I see no point in what they do except one thing: improving the user experience. As a technical user, this is not a big deal for me. As I'll point in a moment, I got some bad impressions from my first experience with Apple product development.

iPhone Operating System

[The iPhone Operating System](#) has derived many features from the Mac OS X and of course, has altered many of them or has introduced new features appropriate for a mobile device, so those who have written programs for the Macintosh operating system should have an easier job learning the process of building an application for the iPhone platform.

The iPhone operating system can be considered as a hierarchical multi-level stack of technologies included in a single platform. This consists of four main layers from bottom to top:

Core OS Core Services Media Cocoa Touch

The first two levels are fundamental layers that provide more basic functionalities and features, services, and interfaces such as APIs to work with files, low-level data types, network sockets, etc. These programming interfaces are mostly provided in C programming language.

The third level, media, provides several APIs to work with graphics, audio, and video, and combines C and Objective-C languages in its interfaces.

Finally, the upper level, Cocoa Touch, which is basically the main layer that programmers deal with is a

higher level API provided in Objective-C to let you apply many of the common tasks in your code.

Typically, an iPhone application is written on top of the Cocoa Touch layer but if you can't find relevant APIs for your work, you may dig deeper and use APIs from the lower layers.

Software Development Technologies

Building an application for the iPhone platform can be done using several approaches. The default approach, as many of you already know, is applying Objective-C and other development tools provided by Apple. This is used by most of the applications written and released on AppStore.

But the other approach that may be familiar to the fellow .NET developers is [MonoTouch](#), a technology released as a part of [the Mono project](#) to let .NET developers apply their knowledge in the .NET development to build applications for the iPhone platform. Basically, MonoTouch binds .NET APIs to native APIs that I described above. It also has a good integration with Xcode IDE that I'll describe later in this post.

My dear friend, [Wallace McClure](#) has written [a Wrox Blox](#) about this technology that is worthwhile to read for everyone interested in using this technology.

Unfortunately, MonoTouch is a commercial product and despite our interest to use it for our project, the pricing wasn't fair for our educational purposes so we ended up with the default approach (using Objective-C).

Objective-C

[Objective-C](#) is a reflective and Object-Orientated derivation of the C programming language that adds a messaging style to the C language.

Objective-C is mostly known for its wide usage in Apple Mac OS X and iPhone development. If you're going to develop for Apple products, you definitely need to know this language in a very good level.

Knowing the C programming language and being familiar with the Object-Oriented concepts, you have a straightforward way to learn Objective-C syntax and style of programming.

Apple developer guide has [a good introduction](#) to this language that can give you a basic idea of the language.

Development Process

The development process of an application for the iPhone platform is very similar to programming for many other mobile platforms like Windows Mobile or Android, but some legal phases to get permissions from Apple add to its complexity that in my opinion seem to be a very annoying part of development process for iPhone platform.

Following some steps, you need to sign up on Apple website and submit your generated local certificates to get a Provisioning Profile (after getting approved by Apple) to use on your machine.

As an academic project, we have access to Apple University Program and had an easier process to get our profiles approved here. We have received a set of iPod Touch devices from the university to use for this project and test our code.

After having the Provisioning Profile, and registering your devices on the Apple website, you can start building applications using the below-mentioned tools. After building your applications, you can test them on a local emulator, and then you can deploy this application to your device as well.

[Here](#) is a simple example that can get you started with the programming style for the iPhone platform.

Development Tools

Another major limitation of development for the iPhone platform is that it mandates you to use a real Mac machine to write your code and deploy them to your devices. So whether you use the default technologies or

MonoTouch, you need to run everything on a Mac machine (and you can't use a Virtual Machine neither).

Having this important requirement, you can use a rich set of development tools provided for the iPhone development. The most common IDE to use for Mac OS X and iPhone development is [Xcode](#).

You can also use the [Interface Builder](#) to put a rich set of controls and user interface elements for Cocoa on your application surface.

[Here](#) you can see a good description of the main development tools you can use for iPhone development.

<http://nayeri.net/getting-started-with-iphone-development>

Merry Christmas

Time is accelerating every year and it's really difficult for me to believe that another year is going to end. I think that it was yesterday that I published [the post for Christmas 2008](#)!

It's Christmas time and for the first time in my life I'm in a country where it's celebrated by the people and it's been a very interesting experience for me so far.

I hope everyone a Merry Christmas and wish that you have great holidays that will be followed by a better year. From my perspective, 2009 appeared to be a better year for the whole world than 2008 and I hope that 2010 continues the progress.

As of myself, 2009 was much better than the last 4 years preceding it. Although it didn't have an amazing beginning and even had a frustrating middle, but the last few months were much better and could erase many bad memories for me.

After [arriving in the US](#), I'm still in Austin but thank to [Mehrdad](#), I've been able to complete many of the initial tasks and hopefully, I'll move in to my apartment in San Antonio during the next week.

Speaking of my status, I've had a good passion to publish some technical posts in the past month or so but as I was relocating, I didn't have a Windows machine to write any code, and also didn't keep up with the news. Anyway, unlike 2009, I hope I can write more in 2010.

<http://nayeri.net/merry-christmas>

So Long Tehran and Thanks for All the Fish

Or it could be as simple as "Hello America" but it's hard to forget the past good days in where I grew up and found myself.

In today's web, thank to Twitter and Facebook, cats don't live inside the bags anymore, so it didn't take a long while for that cat to get out of that bag. Yes, the cat is out the bag and as [Jayson](#), my dear friend, [has written](#) on his blog, I'm already here in the US!

I didn't have any intention to write about it here on this blog, and was trying to hide everything in regards to these matters from the public, but I thought that I have to, at least to thank some great friends that helped me overcome arduous problems in the hardest process in the world. Yes, getting to the US for an Iranian citizen is so hard and we owe that to our governments. The second hardest thing in the world is getting a Ph.D. in the US and I'm going to do that, too!

I had a very long and tiring journey from Tehran to Doha then to Houston, and finally to Austin that took 28 hours, and I arrived on December 8. I've been here staying with [Mehrdad](#), my very great friend who was such an influential person in taking me here, until I recover from the jet lags. Shortly, I'll move to San Antonio to settle in there to begin my doctorate studies in Computer Science by concentrating on Programming Languages and Compilers design. I'll be there for the next few years until I earn my Ph.D. and I hope that I can earn it as soon as possible.

To be honest, my intention to get back to the academy started a short while after [graduating](#) from the university in 2006, and strengthened during the time. Older readers would remember how happy I was when I got out, but soon after going out to the society and working space, and involving in the military service, I realized that this life is not my type and I'm supposed to do something else. Since then (almost for 3.5 years) I had a very tiring, arduous, and sometimes frustrating process to come here to the US for my graduate studies, but thank to all the help and support from some friends like [Mehrdad Ebrahimi](#) and [Kouroush Ardestani](#) ([Robin Pars](#)), I finally made it!

As everyone knows, the past few years of my life were such a mess. Serving in the armies was a total shit and the next 1.5 years had their own difficulties, but recently they turned out to be so helpful to me and making me mature enough for the future events that could melt everyone without strengths. So I should confess that I had some big mistakes about many of those events, and have to apologize for writing some disappointing stuff about them on this blog from time to time. If I go back in the time, I will definitely choose to serve again and pass all the troubles because without the experiences to live under difficult conditions and resist to many problems, I was unable to complete this process. Luckily, I had the chance to have a strong background and resume to go directly for my Ph.D. and save 1-2 years of this wasted time in the military service.

It's hard to write about the motivations to earn a doctorate degree and I don't think that anyone can understand that except those who have engaged in such a program. At some point you notice your talents and what God has put inside you. To be honest, I also realized that it's hard for me to deal and work with ordinary people and it yields complains on my side as many of their actions don't make sense to me. I'm not saying that others are good or bad. I feel that we have different worlds and that's all. Therefore, I was looking to be somewhere and work with some people who are closer to my opinions and viewpoints, and the best place could be the academy.

Besides, I chose to come here to the US for several reasons. The first and foremost one was the unique and world-leading education system that is much stronger when it comes to the Computer Science because almost everything in the computer world is introduced and developed here. I was looking for a real cutting-edge research and it wasn't easy to find it in many countries including Iran (despite the very strong system that we have in there and many of the new generation youth don't know). Furthermore, if I had to choose a third country to be in for a long while, it could be the US because I knew their language, culture,

and many other things much more than any other country in the world, and I proved to be good at communicating with these people mainly because we share some interests specifically the passion to lead and be that number one thing!

All in all, to make the long story short, I switched from doing what everybody does and what the environment forces me to do to something that I'm supposed to do, and it's my purpose to be in this world. Unfortunately, this costed me too much but after all, I'm happy with everything happened. I could start my Ph.D. at 22 or 23 but still I'm so young and maybe one of the youngest Ph.D. students here at 25 specifically as I met many people studying at college at my age even older.

The upcoming years are hard as earning a Ph.D. is, naturally. It's harder for me because I'm switching from Mathematics to Computer Science, and am going to live in a new environment missing my family, and that's usually difficult for Persians as we are very emotional.

Leaving my homeland was really hard, and it was a matter of choosing between losing some stuff to have new achievements when I was doing that. I won't be there for the coming years but I hope that our people can get their share of liberty as they proved to deserve it. Believe me or not, there aren't many nations in the world who truly care about their freedom to sacrifice their bloods for it. I think that all the recent events showed the huge difference between the Iranian nation and the regime, and I hope that Iranians can get a better assistanship from the outside world rather than some stupid sanctions that are eventually weakening them and empowering the dictatorship.

In the end I have to thank everyone who helped me in any way in this process again, and thank the American nation, especially these warm Texans, in advance for hosting and enduring this monkey for the next few years! I hope that I can contribute back to this country to assist them in accelerating their progress that has slowed down in the past decade. The least I can do is opening up new job opportunities and a leading position for them with my research even though I've proven to be able to do more.

Humankind can go and get to any point that he imagines. It just mandates him to really want it, and put enough effort into that, and I've experienced that in my life several times. I never earned anything in one night (even disregarded anything came to me that way) and achieved everything by hard-work and it's been the main reason I could continue my progress in the time.

<http://nayeri.net/so-long-tehran-and-thanks-for-all-the-fish>

Twitter in a Dead End

These days it's hard to find someone who doesn't know about [Twitter](#) as a social networking status updating site mainly founded in 2007 with the very simple and basic idea to let people share their status on the web with 140 characters and follow others' updates.

I was among the first group of users who appeared on Twitter and have been continually tweeting for almost three years to now, so I've been able to witness the progress of the site and the way it evolved.

The idea behind this service was very simple that had been tried by some other sites before, but Twitter was able to gather a reasonable number of known people to incorporate because it had a very simple and basic structure, and was founded by some of the well-known names on the web. In this while many rivals have appeared to challenge this success but none of them could go so far.

As many other people have stated already, in my opinion over half of the Twitter's success was related to its simple idea and implementation that despite the technical issues in all these years, has helped it survive and get more users. This simple core allowed developers and other companies to build services and sites to complement the basic structure and make it suitable for specific applications, and inspire it as a general platform for content publishing for various applications; therefore, a diverse group of users benefited from the service whether for personal updates, entertaining stuff, technical matters, or anything else.

Like any other service or site, Twitter had its own progress. From a perspective, there were two waves of progress for Twitter: the first wave was the appearance of new users as individuals, companies, or authorities, and the second wave was the addition of new features by Twitter leadership to improve it or by adjunct services or client applications to simplify a task for the users. Most of the features added to the core service/site were inspired from these third party tools and projects that could gain a widespread popularity among users.

Both these waves had a good progress and quickly took Twitter from a very lower point to somewhere that is now a de facto standard of online communication among many individuals, colleagues, companies, media, and politicians. I can exemplify this by pointing to the main role of Twitter in post-election events in Iran or its usage for communication between the American nation during the last presidential election. Likewise, it's used widely to share the news, opinions, and interesting updates about some events such as Microsoft, Apple, or Google conferences.

On the other hand, as I [had written](#) before, Twitter and micro-blogging came to challenge blogging and had their obvious impact on the frequency and quality of blogging and online publishing for almost everybody around the world.

However, despite all these matters, recently I came to the conclusion that Twitter has begun to worsen, and is slowly losing its active role on the web, and that's the reason for me to write this blog post.

I think that there are three main threats challenging the future success of Twitter:

Development of the service and appearance of more users, specifically ordinary people, on the site without considering a good mechanism to maintain the privacy and quality. Leaving the simple nature of the service by implementing some new features by Twitter leadership/developers. The advent of Twitter spammers and spamming on the site specifically in the trending topics. The Number of Users - A Double-Edged Sword
The first and foremost threat for Twitter is the number of users! With almost no exception all the social networking sites founded during the Web 2.0 era had a similar progress starting from a lower point, evolving with a good speed by adding good features and attracting users, and stopping at a higher point, then slowly losing the quality. The only exception that I can note is Facebook and I'll explain why.

Apparently the number of users is a very important factor in the success of a social networking site (even though I personally consider the level of education and sophistication of the users as the primary factor), so

all these social networking sites compete to get more and more users on board. This isn't necessarily bad because it brings a wider community of people to contribute and improve the level of communication and information sharing. But if not handled in a good way, this also has its own drawbacks that even can drive the whole community to a bad point.

Usually social networking sites begin with a small group of users that are often technical people and community leaders. These users are innovative, sophisticated, and well-aware of the culture of online life, so they communicate wisely and take the site to a better point. When new users, that are mostly more ordinary people, come to such sites, less or more, they lack that level of sophistication and awareness and it can have a huge impact on the quality of the content. If the site administrators don't handle this wisely, it can cause the growth of mushroom-like groups of users who disturb the site. I think that everybody can notice this on many of the social networking sites such as the first-wave sites like Orkut or middle-wave communities like MySpace.

I believe that the best way to prevent such issues on a social networking site is to isolate users to some degree, so they let each other communicate if and only if they're interested to do so. In other words, a powerful privacy management mechanism can maintain the independence of users on a site, so even if someone is bothersome, he or she bothers his or her friends/contacts only. I believe that this is the main reason for Facebook that has let it continue its growth and success. Facebook allows all users to have full control on their privacy settings and share certain information with their friends.

In my experience (that is easily provable by reviewing the history of social networking sites) the number of users is a double-edged sword that if controlled and managed, can help a site succeed, otherwise, it becomes the main weapon against the site, itself!

Unfortunately, Twitter lacks a good mechanism to allow its users to maintain their privacy. While I'm a big advocate of simplicity in the core service and site (as I'll explain in a moment), I strongly believe that the privacy is the only area that mandates a lot of features even complex features. Being able to make something public or private/protected is not enough! Here I note that obviously the type of privacy settings would be different from Facebook or other sites. For instance, I always wanted to be able to hide certain people that I'm following from the public eye to prevent some people from stalking (thank to [Steven Smith](#) for introducing me to this term).

Recently I protected my own Twitter account and removed many of my followers for this specific reason. Since then I've been rejecting most of the follower requests to keep a small but active group of followers for myself who aren't possibly a threat to my privacy.

Complexity - Back to 90's

In the introduction text I stated that simplicity was the key factor for the growth and success of Twitter but in the past couple of months they have forgotten this principle and have been adding some unnecessary features (in my opinion these features are redundant for the main site) with a very bad implementation. Two notable features are the Lists and the Retweet button. Not only these features aren't necessary for the site, but also they're implemented in a very ambiguous and complex way, so still I have difficulties in understanding which is which with each feature. In my honest opinion Twitter hasn't made things complex. It has taken an step further and has returned to 90's days of poor user experience needing a guide to teach each and every user what to do!

I'd like Twitter to be as simple as it was before, with a textbox that lets me write 140 characters of text only, and I'd like this service to be complemented by a rich API that allows me to expand this basic idea for anything I need. If I want to Retweet, then I can go and download from hundreds of clients that have much simpler and richer Retweeting features. If I want to categorize my contacts, I can go and use one of those clients or online services.

Basically, it doesn't make sense to see Twitter leadership focusing on these redundant features while there

isn't a single day that all users don't see those funny errors. Why don't they concentrate on scaling up the service to handle all the requests rather than adding new features? Isn't it obvious that the former one has more priority?

Spamming - The Old Enemy of Content

The last main threat to the success of Twitter is our old friend, spamming. This one is the natural cause of this type of content publishing. There are various types of spam and tweeting has its own spam as short text messaging does. On Twitter there are two general types of spamming: one that follows many users to receive some mutual followers to publish advertisements for them, and another that uses the popular trending topics to promote something in front of thousands of eyes. These spam updates are slowly growing and disturbing the relevance of content.

The problem with Twitter is that it still doesn't have a very effective mechanism to block these spammers and filter the content and solely relies on user reports. Of course, one technical difficulty is the length of Twitter messages that are short enough to prevent automated anti-spam methods from filtering them.

Conclusion

All in all, I haven't liked Twitter recently as I used to in the past. Different people have different expectations from the people they follow and from the general content. Some people are interested in technical discussions, while some others are interested to read technical news and links. There are some users who like to see personal status updates, but there are others who like a combination of different types. There are even people who just want entertaining images, videos, and jokes!

I'm not very interested in those personal status updates even though I don't reject them all, but there are some users who salt it!

Besides, in my understanding the usefulness of live event reports is lowered on Twitter as there are more people who repeat something over and over. One instance was this recent PDC09 event where I was witnessing something repeated by many people. Believe me, it becomes boring at some degree! I'm lucky to follow a very limited group of community leaders only!

To summarize my discussion I have to say that Twitter needs to revisit its new strategies, otherwise I can't imagine a better future for this site. You, as dear reader, may argue this and oppose to my statements here but usually such predictions rely on experience, and only time can prove what is right and what is wrong.

As of [my own Twitter account](#), I've been thinking about stopping it and continuing everything on [my Facebook profile](#) where I've been so active recently. I think that tweeting doesn't have much value for this world while I and many other people can save our time and effort to use them for more beneficial tasks.

<http://nayeri.net/twitter-in-a-dead-end>
