1-1 Why do Rails routes map to controller actions but not model actions or views?

Since by assigning the routes to controller, controller is able to know which request should do which query to model. After getting the data from model, controller can show the corresponding data to views according to the routes.

1-2

Joel, the author try to convey an idea of “The law of Leaky Abstractions”. He start from telling an analogy of TCP and IP mechanism. He is talking about TCP like an express way from point A to point B, but doesn’t guarantee you will arrive, you will probably be directed some other places.

Anyway the original mechanism will never be that easy as we think.

The main idea of “leaky abstraction” is, for some mechanism, programmers or lecturers want to simplify the explanation or usage of it. But somehow in the future, the users will find some errors or issues, which force them to fix or to understand from the original idea of that specific mechanism.

Also, Joel provides some practical example on leaky abstraction.

. Some SQL query syntax might be much faster, but using that sometimes might “leak”, which is, make some error and we still need to go back to original, slower way.

. Some application allows use can remotely reach the files in library as if those files are local, sometimes this mechanism will become slower or even stop. Programmer need to fix when errors happen.

. The basic concatenate of string in C++.

. Give an analogy of drivers can’t drive too fast because of hydroplaning.

To answer the question in assignment:

a. I believe leaky abstraction will also apply on MVC model. Since MVC split the webpage to three main parts. I believe in the future we must meet some errors, find that MVC isn’t that universal to solve it. We probably need to code everything from scratch without MVC concept, or have to understand the real concept behind the scene of MVC.
b. I think, yes. Since Ruby is very easy and user friendly, obviously there must a lot of processing mechanism behind the scene that we don’t know. For Rails API, yes. We met a couple problems when we applied some gem files on internet, and can’t find existing solution on the internet. At this point, we need to mock

the API from the scratch, or need to fully understanding how that gem file work. I believe this phenomenon is so-called leaky abstraction.

c. Agree. But actually I didn't met this issue yet, since I didn't use that difficult coding skill yet.

1-3

Steve, the author in this article, describe the importance of using verbs in daily life as well as in programming world.

Steve begin this article from telling a story: there are a lot of language lands, and citizens of each lands have different rules.

For example, the king of Java island believe verb is not safe, so he use corresponding noun to replace all of the verb.

However, this is not the case in other many kingdoms. Like in C-land, JavaScript land, and most important, the Ruby-land, their citizens can use noun with verb, which make the life a lot easier. They can describe or apply an action on an object in an appropriate way.

Also, in those verb-using-permitted lands, they don't need to use a lot of terms to wrap the noun. The can announce a verb directly, and don't need a following noun as an object.

After this story, Steve come back to the real programming world. He mention that C++ give the users verbs other than nouns. However, Java is not.

The worse thing is Java insist that the user need to use class: When user come up even a very simple idea, we need a lot of classes to wrap that idea, which makes thing complex.

Answer the question in the assignment:

a. Ruby- what the Steve said doesn't 100% fit my experience. There is no doubts Ruby is much easier (in terms of syntax) than Java. But to my understanding even Ruby doesn't use a lot "verb". That is true in Ruby world we don't need to define a lot of classes, but we still have objects, methods, instance variables, etc. It is still an OOL. Therefore, I treat Ruby as a simpler version of Java.
b. Partially agree with this article. It's probably true that we can't use "verb" in Java, but Java is not that bad because of this, it is still powerful. But apparently the languages which allow user to use verb is more straightforward.