Common Simulink library.
April 28, 2016
Prepared by: Vijay Kotnoor.
Problem Statement

“Re-usability of simulink library”

Q: How did start?
A: It started with a question “Why am using a different library block(in terms of implementation ) to do the same thing when am working on a similar product?”

Q: What answers did we get to the question “Why am using a different library block(in terms of implementation ) to do the same thing when am working on a similar product?”

A1: Oh we use different Matlab/Simulink versions.
A2: That’s seems a better solution but we don’t want to change it now.
A3: We did not know this was there in that product we made our own.
A4: Our library has a small difference than the one you are speaking, so we could not keep it common.

Each of the answer implicitly have a problem which needs to be solved if we have to move to a common library.
Why do want to solve have a common library?

• Having a common library will reduce the effort taken in the software development.

• we can use the most efficient library blocks.

• From verification point of view we just need to get the library verified and all the projects can reuse the same stuff for all the projects.

• Benefits Improved efficiency of the code.
  ■ Reduced effort in the software development.
  ■ Reduced effort in the software Verification.
  ■ Reduced training cost when moving resources.
Approach used to solve problem: Answer 1

Let's start with the first answer
“**A1: Oh we use different Matlab/Simulink versions.**”

Q: Has somebody faced the same problem.
A: Yes there was a team which was using different Matlab/Simulink versions.

Q: How did they solve it.
A: They have different libraries maintained for all the versions of the Matlab/Simulink libraries. Startup m script was used to load the appropriate library for the Matlab/Simulink version which was being used.

Q: Is this not a overkill to maintain so many libraries?
A: No mostly upgraded the libraries with to newer version of Matlab/Simulink and they were not facing any major problem doing it.
Approach used to solve problem: Answer 2

A2: “That’s seems a better solution but we don’t want to change it now.”

Q: What do you mean by now.
A: We are at the end of the project so don’t want to change.

Q: Would be using it in future?
A: Depends on the amount of change we have to do existing models.

Q: If we also provide solutions to new problems which have been solved else where and are not part of your product now will you be interested.
A: We might be interested in them.

So we found out what features implementations are new. But we have couple more things solve.
Approach used to solve problem: Answer 2

• The new solution were project specific and not part of the library.

• We also had to start looking at uniqueness of the solutions which were hidden in the model and not part of the libraries and move them to a common library.

• This just increased the scope of the what we set out to do.

• But then this helps us re-use solutions which are unique, else we would have to redo the same stuff to arrive to a solution and that might be completely different.

• Its not the commonality but the uniqueness which adds more value to the library.
Approach used to solve problem: Answer 3

A4: “We did not know this was there in that product we made our own.”

Q: What do you think is needed so that you know it’s there.
A: A place where we can go look at what’s available.

Q: Don’t you think you might have to look at too many places.
A: Yes that would be a difficult task.

Q: Having them in library would make it easy for you to look at one place.
A: That would make our job easier.
Approach used to solve problem: Answer 4

A5: “Our library has a small difference than the one you are speaking, so we could not keep it common.”

Q: Would be okay given a option (a pop to set parameters) so that we can keep the model same and have both of them implemented in a single model.
A: Yes that should be okay

• Most of the libraries were changed so that when ever a library block is dragged to the model it will come up with a pop with option of what you want to so. And change the implementation like wise with small difference.

• This helps keep the blocks more generic and can be used by many people with the differences they want.
Approach followed.

• Check for commonality or uniqueness in the blocks.

• Look into different approaches used by different display programs.

• Compare blocks from different display programs.

• Try and write a backend .m script so as to have single point access to similar blocks

• Make the blocks more dynamic and take inputs when dragging the blocks to the model. (Using a pop up menu)
Results achieved:

- A total of around 25 library blocks were created/modified to be part of common Simulink.

- Reduced the number of library blocks by giving option in the pop-up. They either had to be hardcoded in the model earlier or had to have different library blocks. More than hundred blocks were reduced to be only 25.

- POC was done by using these block sin models and generating the code and doing simulation.

- Some new ways were found to implement things which needed a lot manual work. These were completely new and more effective.

- Many algorithms which were hidden in different models and libraries in various projects were put together in one place.

- A common solution was found for recurring problems.
Future Challenges

• Start looking at the more opportunities where we can use similar concept and work towards standardizing the libraries.

• Start targeting any new program which is starting from now and work with them and present them the use case for using these library.

• Coordinate and help any team using the library and resolve any queries.
THANK YOU

Questions??