

The limitation of my application is the states of 'Not in Grammar' and 'No Voice'. There are many similar code blocks that I think I will improve them to be shorter in the future. Also, I think the grammars should be larger because it is limited that the robot only knows a few names and times. Therefore, if we utilize a huge amount of data of person name and so on, the robot will be robust, in my opinion.

When developing my app, I met a problem with using the 'cond' or 'guard' because there are many methods of using them in official documents. For example, guard implementations can be quickly prototyped by specifying the guard cond function inline, directly in the machine config:

```
// ...
SEARCH: {
  target: 'searching',
  cond: (context, event) => context.canSearch && event.query && event.query.length > 0
}
// ...
```

However, when I use 'cond' and 'target', I didn't find the reason why it failed to jump to next state. Finally, I fixed this problem by using 'guard'.

Another challenge I think is where to write the automatically start code. At first, I wrote the 'setTimeout' function in 'element.addEventListener', but it didn't work. Then, I made all 'dmActor' start automatically using the following code. Finally, the 'dmActor' can start without clicking the button in html.

```
element.addEventListener("click",
() => {
  setTimeout(function(){
    dmActor.send({ type: "CLICK" });}, 100)
});
setTimeout(function(){
  dmActor.send({ type: "CLICK" });}, 1000);
setTimeout(function(){

  dmActor.getSnapshot().context.ssRef.subscribe((snapshot) => {
    element.innerHTML = `${snapshot.value.AsrTtsManager.Ready}`;
  });

}, 10)
```

Last error, I feel most confused is the usage of local variable and global variable here. Since, there are lots of nested structure in 'dm.js', I found it is difficult to change the global variable if I defined it at the beginning of my code. For example, when I used actions, I couldn't change the global variable by the 'action: assign' expression. Even though I tried the feature of 'meta' block, which we can access the information in meta block, it still didn't work. I need to learn more about variable reference in nested code block. I haven't found the solution for this question.

