# Action Items and Important Ideas:

1. Work on Sirius editor to provide a self-contained plugin to enable statechart model development (diagrams) and generate a SCXML representation. Also allow edits to the SCXML to propagate to the diagram as well.
2. SCXML plugin and paper on the current work. This will focus on enabling refinement while supporting and SCXML run to completion semantics.
3. Exploring the generation of different refinement model combinations. The idea here is to keep each of these refinement levels in a SCXML representation to enable leveraging the plugins already developed. To allow for multiple refinement models the SCXML model provided by the developer must have the following:
   1. No explicit refinement level information
   2. The data model with invariant and data type information is defined within each state at the level of hierarchy at which is introduced.
   3. Each guard and action for a transition is defined separately (no use of the condition attribute)

Additional logic will be required to identify which guards and actions correspond to which refinement level depending on the data defined in the data model

1. Use the SCXML model and generate an Event-B representation of the model using clocks. This approach should guarantee satisfying refinement proof obligations
2. Use the SCXML model and exploit the q tool to create an Argos flatten version of the model. Then use the resulting intermediate representation to generate and Event-B representation of the Argos model. Things that we need for this:
   1. Open source q tool
   2. Generate some examples of the type of properties we want to proof
   3. Explore the use of gluing invariants to relate Event-B model properties to the changed Argos representation.
3. Ask Michael to describe his suggestion for developing models that are synchronous and only introducing asynchronous in later refinements.