The Problem:

Creating a data structure that had the following capabilities. Navigate along a given path of actions and decisions, Multi-Threaded processing, partial self-driving, and easy maintenance.

The Solution:

The structure I have created has two building blocks, nodes and decisions. Each node has three primary pieces of metadata, a question, node event, and listening type. A question is what the node needs to know before moving forward, there are two types of questions. One that gets asked to the user directly searching for answer, the other is a comment on the node action, which does not require human intervention. The listening type ties into the LingPipe natural language processing model, which will be used to search the tags in the resulting sentence driven by the question. A node event is used when human action is not required, this allows for a self-driving structure. An example of this is searching the vitals database for a particular variable, then passing that variable into a decision structure.

The second building block decision has three primary fields; answer, destination node and event. The answer is what is being looked for by the node, this can be in a number of forms, ranging from an exact string of text to FOUND to a variable range. Destination node is the node that the decision will move to after the given event has been fired. This particular field brings the data structure full circle; nodes have decisions which leads to nodes. The event is what happens when that particular decision has been reached.

Once those building blocks have been established they are sorted into clusters of like-minded nodes which are kept separate from each other logically. This separation into different clusters allows for multi-threading. Allowing each cluster to act independent of other clusters allows for multiple clusters to run concurrently.

Continued Work:

The remaining problem is easy maintenance. Currently maintenance of this data structure is complex and needs to be performed very carefully and can quickly become unruly as the structure grows. My proposed solution is creating a program that can maintain the database of nodes and decisions.