The Problem: Several other abstracts (Ingham, Walcott, Wehowsky, Stedl, Williams) present the legion benefits and applications of the model-based programming approach to developing autonomous embedded systems. Introducing planning, mode estimation, and deduction to the runtime system enables the specification of embedded systems at a higher level of abstraction than was possible without these tools. However, model development and testing is still quite labor intensive and error prone.

Motivation: The higher level of abstraction available to the model-based programmer enhances productivity and correctness, but can also make it less clear what is actually happening at runtime. To support both the development and debugging of model-based programs, we have integrated some graphical modeling tools into the RMPL [1] development process.

Approach: Using the metamodeling tool GME 2000 [2], we have generated a modeling tool for RMPL control programs. The GME-based GUI can be used to directly specify RMPL control programs. When the program is ready to be executed, there is an export-to-hca function that generates the compiled form of an hierarchical composite automaton (HCA) to which RMPL control programs compile.

It is also possible to develop RMPL programs textually and export them to XML, which is in turn able to be imported into the GME-based RMPL modeling GUI.

For debugging and experimenting with the RMPL program, it is possible to execute the compiled RMPL model and have the GME-based GUI highlight which states in the HCA are marked as execution proceeds. This can be an invaluable aid in debugging a model-based application. We also have integrated the Helios tool from Johns Hopkins APL to visualize a plant model.

Figure 1 shows a GME model during execution, as well as the Helios tool displaying active states in the plant model. The lower right is the Helios display of the plant model, and the upper left is the GME display of the control program. The darker states in the control program are marked in the underlying automaton.

Future Work: In addition to developing visual programming and debugging tools for model-based programming, we are exploring the idea of applying model-based programming techniques to “general purpose” programming (as opposed to autonomous embedded systems programming). The idea is that programmers can specify their intent and alternatives during the design process, and these specifications may play an active role at runtime, rather than being mere artifacts of the design process.

Research Support: This research has been supported by a grant from DARPA, under the Model-Based Integration of Embedded Software program, contract number F33615-00-C-1702.
Figure 1: An RMPL Program Running under GME and Helios

References:
