
Type of paper: This paper describes a radical solution.

Correlation to the real-world setting: cites former studies that programmers occasionally violate the group’s consensus on how modules should be connected.

What is the hypothesis: There are two distinct activities in programming: writing modules and connecting these modules to form larger programs. These activities need separate languages. Current programming languages do adequately support the module (inter/dis)connection problem.

Validation: They examine an existing system to glean its structure. This is difficult, and the original programmer didn’t understand the structure to begin with (proof that the problem exists). They create a module interconnection language and apply it to the same program. Then the authors fall back to persuasion to argue that the MIL is a good thing.

Result: Identification of module interconnection as a problem, definition of MIL 75, and description of a technique to check the language.

Do you believe the result, and why? I’m persuaded that the problem exists, but I want to see a cost/benefit analysis of the technique. Could the time spent on the module interconnection language be better spent somewhere else?