

Chris Cargile - January 30, 2014

CSCI658 - Testing and Maintenance

5.1) Describe how requirements of model compactness, predictiveness, semantic meaningfulness & sufficient generality apply to approaches for modularizing models of programs, supported through examples where possible:

By using various approaches to modularizing models of programs, the (4) modelling requirements can be taken into consideration so as to make models of a larger system conceptually encompassing of the general properties that describe a system (compact/concise) and its outcomes (predictive), in such a way as to convey key concepts and good vs. bad outcomes, and provides enough detail to afford insight into specific areas for improvement (semantically meaningful). All the while, said modelling approaches should be delicate in their representations of the domain so as to remain applicable in similar settings ('keeping general'). For example, using basic blocks in a control flow graph ascribes to the notion of compactness, which assures comprehensibility. In developing models, tradeoffs regarding the level of specificity to which control flow is depicted, along with the allowance of excluding certain implicit control flow edges (and thus some paths) can be an indication of the margin for generalization that is tolerated by the model.

5.2) If models are sometimes analyzed to build programs, but sometimes they are built to model a program that is already built, then why bother with them at all; and why not just merge the design considerations in the implementations of the latter system to be developed?

As a model describes the behavior in more general terms, it can not only convey the broad meaning and details that the implementation affords, but they may provide easier means for realizing commonalities from one domain to another. For example, modelling a control flow analysis for conditional statements, may give way to a realization that visiting all the possible branching paths in a program is the same as recursively generating a minimal-spanning tree or some similar type of data structure. Therefore, models *may* serve as a visual, higher-level counterpart to the initial analyses conducted prior to implementing a software program.