Formal Specification and Requirements Modeling: Specifying the Mode Logic of a Flight Guidance System

Gary Daugherty, Rockwell Collins, gwdaughe@collins.rockwell.com

Incomplete, ambiguous, and rapidly changing requirements can have a profound impact on the quality and cost of software development. This is particularly true of reactive systems, with complex mode transitions, such as the mode logic of a Flight Guidance System for a General Aviation Aircraft [1]. Based on our experience in modeling the requirements for such systems, it appears that:

High level specifications written from the customer's perspective can be difficult to translate into formal or semi-formal requirements models written at a more detailed level for software developers. Doing so, however, serves as a useful step in the refinement of requirements into code.

The ability to map between these two views of the system, however, is important, since customers and users are often unwilling to deal with the level of detail that appears in most formal and semi-formal models.

It is important to use models (such as hierarchical finite-state machines) that correspond closely to the domain expert's mental image of the FGS mode logic.

Graphical representations are valuable only if we can avoid unnecessary clutter – for the FGS model, the use of a transition bus, rather than individual transitions between states turned a virtually unreadable diagram into a valuable visual aid.

Features that help us avoid a combinatorial explosion in the cases described by the specification (such as support for concurrent modes in state-machines) are essential.

Knowledge of the problem domain and appropriate modeling methods are more important than tool support when it comes to building the requirements model.

Although simple syntactic checkers, editors, and document preparation facilities can be very helpful, they must support an appropriate modeling notation and methods. Tools that fail to do so (e.g., tools that fail to support hierarchical and concurrent states) are more of a hindrance than a help in developing a readable specification.

The real value of a requirements model depends upon the definition of an associated, precise semantics. This permits the execution and visualization of the requirements by the customer, the automation of consistency and completeness checks, the automation of safety analysis, support for the proof of key system properties (such as safety, liveness, and timing), automatic generation of test cases, and the generation of code from models.

Although the modeling methodology may emphasize planning for change, this is not the same as planning for a product family development. The mode logic for the FGS requirements model required a complete restructuring to support variations among product familiers.

References

[1] Steven P. Miller, "Specifying the mode logic of a flight guidance system in CoRE and SCR", *Proceedings of the second workshop on Formal methods in software practice*, p.44-53, March 04-05, 1998, Clearwater Beach, Florida, United States