Abstract

The productivity and dependability challenges we face in software engineering remain significant. Software has become the pacing item and an unexpectedly large expense in the development of many engineered systems. Dealing with this situation will require a radical change in the way we build software, and one candidate to form the basis of that change is model-based development. Model-based development relies on the creation by application experts of models of the required systems and synthesis of the software from the model. This approach brings many benefits but also a surprising number of challenges. Success to date with model-based development has been considerable.

1. Introduction

The demand for software continues and with it the demand for software engineers. Software has become the pacing item and an unexpectedly large expense in the development of many engineered systems. The quality of the software being developed remains, in many cases, quite mediocre, and the consequences of that mediocrity impact us all in many ways, from serious security failures to the loss of availability of services upon which we depend.

There is no question that things have improved considerably thanks to the many novel ideas that have emerged from research programs. Despite these advances, dealing with this situation will require a radical change in the way we build software, and one candidate to form the basis of that change is model-based development.

2. Model-based Development

Model-based development starts with the creation by application experts of models of the systems that they require. Tools allow models to be edited and analyzed, and in some cases models can be “executed”. Once the model is complete, the tools synthesize the final implementation from the model. The models are built using a variety of formal notations but graphic notations are common. Two examples are SCADE and Simulink. Both are designed primarily for defining models of control systems as are found in embedded applications. Although this is a narrow domain of applicability, the penetration of these techniques in practice is very high.

3. Benefits And Challenges

Code synthesis is not new, but model-based development is a radical change in the way that software is built because of the central involvement of application experts. Claimed advantages include less development effort, easier validation, independence of application language, and no defects arising from human development.

Despite these advantages, current instantiations of model-based development are raising fundamental questions about the notations being used, about the established software artifacts, and about software-development processes. The notations are describing a computation yet many lack support for exceptions, concurrency, and real-time processing. It is not clear whether the model should be viewed as a specification, and both verification and validation become blurred concepts. The issues become especially significant where software is being produced that has to be certified by a regulating agency since existing standards are stated in terms of established terminology and concepts.

4. Conclusion

Model-based development offers many advantages over traditional development but significant technical questions remain. The advantages suggest that it will be a powerful force in future software engineering.