Integration, the Price of Success
Extended Abstract

J. Sifakis

Verimag, Centre Equation, 2 rue Vignate, Gières, France

Formal methods and their acceptance

It is generally recognized that formal techniques have a limited impact on current industrial practice. Several reasons often had been advocated to explain this fact: youth of the discipline, intrinsic limitations due to complexity and undecidability, lack of trained practitioners and engineers. All these are factors limiting the application of formal techniques. However, a reason that is less frequently suggested is the relevance of our contribution to current industrial practice. Clearly, Informatics is a scientific discipline with its own evolution laws and proper objectives. However, as an experimental discipline it should find inspiration and validation in applications whose development is also driven by external needs, technologic, economic and ultimately social. The recognition and the success of our discipline is intimately related to the capability to address problems raised by the fast evolving practice.

Many theoretical results, concepts and methods have been integrated in current engineering practice so perfectly that their use is completely transparent: automata and formal languages theory in compiler technology, or more recently, circuit verification by model-checking. On the other hand, it is clear that most of the existing theoretical results will not find a direct application in the foreseeable future while at the same time software and systems engineering techniques develop on a more empirical and technology driven basis. It is remarkable that pessimistic predictions about limitations of the current industrial practice to produce reliable software and systems, have been falsified. In fact, the rapid increase of complexity and the introduction of new technologies arising from the integration of reactive systems satisfying hard real-time requirements, did not result in significant changes of the current practice.

These facts should make us question the direction of our work, in particular, its role and the relevance of its contribution to Informatics. Without abandoning promising theoretical research directions, it is important that a stronger connection with practice is sought for two reasons: to defend Informatics as a basic emerging experimental discipline and more importantly, to validate results, methods and ideas by confrontation with practice.

Research on formal methods has essentially been motivated by the study of formal languages and their semantics including languages for the description of systems (executable languages) and of their properties (usually logical languages). In this context, two important and related classes of methods have been extensively studied. Synthesis methods which allow the computation of

J. Wing, J. Woodcock, J. Davies (Eds.): FM'99, Vol. I, LNCS 1708, pp. 52-55, 1999.

a system satisfying a given property and verification methods which allow the satisfaction of a property by a system to be checked. The effective use of formal techniques by software and system developers requires their integration in existing support environments. There are two major impediments to such an integration. One is the distance between formal languages and languages used by practitioners in different application areas. The other is non scalability of existing formal methods.

Connection with practically used languages

Industrial practice is based on the use of sets of languages covering the development cycle from requirements, to specification, programming and integration. A remarkable fact about the evolution of languages over the last twenty years, is the common use of a few implementation languages and the emergence of executable description and development languages specific to application areas. For each of them, complex technologies with commercially available support tools have been developed. For HW design, there exist CAD tools dedicated to languages such as Verilog and VHDL. For real-time systems, two different technologies emerged, one based on ADA and multi-tasking and one based on synchronous programming. For telecommunications protocols, a complex technology for specification and testing is being developed around standards such as SDL (executable specifications) MSC (service specifications) TTCN (test case description format) defined by ITU. Finally, for distributed systems standards such as CORBA, IDL and UML are promoted due to the concerted action of leading software companies and organizations. Clearly, most of the mentioned formalisms and languages are not impeccable and sometimes do not satisfy elementary rigorousness and optimality requirements. However, their impurity very often hides interesting problems that are crucial for modeling complex systems. The study of executable languages for systems design and modeling as well as their combination in methodologies is becoming the most important research direction in Formal Methods. We identify below some interesting problems:

- Combination in co-design methodologies of hardware description languages with system description languages, such as SDL.
- For real-time systems engineering, the combination of the synchronous approach with asynchronous multi-tasking approaches requires a deep understanding of the underlying models. In the same area, we need methods for combining description of real-time controllers with models of their asynchronous timed environment involving continuous state variables.
- For distributed systems, we need languages with dynamically changing structure, process creation and mobility. Another important problem is the introduction of a "reasonable" notion of time in specification languages such as SDL or UML. Finally, the definition of languages for general distributed systems raises a multitude of questions about optimal combination of different approaches and description styles as illustrated by the definition of UML.

Scalable (partial) methods

Most of the synthesis and verification methods are not tractable for real systems and non trivial properties. Even though in some applications such as circuits or simple real-time controllers, current model checking technology may suffice, in other applications involving complex data and asynchrony available tools are severely limited. Furthermore, it is anticipated that the complexity of systems will increase rapidly while methods are subject to intrinsic complexity or undecidability limitations.

Research on synthesis and verification techniques focused so far on general methods for powerful languages of properties. Prototype tools have been often developed with a concern for generality which is valid in research terms, but is incompatible with efficiency. To get synthesis and verification techniques accepted, another approach would be to develop methods that allow current industrial practice to be improved by respecting tractability criteria. This practically means restriction to very simple classes of properties such as reachability and invariance and preference to partial methods that can be combined smoothly with other more empirical techniques. We identify below some interesting work directions:

- Combination of model-checking and of static analysis techniques with debugging and simulation techniques to validate complex systems. Discovering design errors by model-checking for safety properties does not require an exhaustive exploration of the state space. The use of on-the-fly techniques as well as approximate symbolic analysis techniques such as convergence acceleration and abstraction can drastically help validating complex systems provided they are integrated in support environments.
- Improving testing technology, especially functional testing. Testing is the validation technique used to check experimentally, by application of test cases, that the behavior of a system agrees with some reference property. Testing is a work intensive activity and in many application areas test cases are written by experts from requirements expressed as "test purposes". Test case generation for a given test purpose and a given system to be tested, can be formulated as a synthesis problem. However, for test purposes expressing simple reachability properties efficient automatic test case generation techniques have been developed and are now available in commercial tools. These techniques share a common technology with model-checking and can be drastically improved by symbolic analysis.
- Use of synthesis techniques in design methodologies. Standard engineering practice consists in decomposing the system to be designed into a set of cooperating components. A key problem is the coordination of the components in order to ensure that the global behavior satisfies given properties. In some application areas synthesis techniques can be used to solve the coordination problem by computing a controller that adequately restricts the behavior of the components. We believe that such a combination of synthesis

and design methodologies is particularly interesting for scheduling real-time applications or in general to design hard real-time systems.

Discussion

The effective integration of formal techniques in design methodologies and support environments is a significant criterion of success. For more than three decades formal methods have been developed rather as an alternative to some industrial practice which it was envisaged would disappear. Instead of the expected revolution, we observe a slow integration of those results that improve incrementally a well-established practice. This process is determined by needs and rules that lie mostly outside our area of action. It favors the integration of formal techniques which are the most ripe for transfer: tool supported techniques based on executable languages.

It is necessary that we participate in this process by focusing on new challenging work directions. This implies a deeper understanding of current practice, methodologies and technological results that are at the basis of software and systems engineering. This requires in some cases, critical mass and organization of the effort on a different basis. Our success and credibility depend on our capability to take up the integration challenge.