# A SURVEY ON FORMAL METHODS USING IN SOFTWARE DEVELOPMENT

*Juan You[1], Junquan Li[1], Song Xia[2]*

*[1]Zhengzhou Information Science and Technology Institute,Zhengzhou, Henan, China*
*[2]Hefei Electronic Engineering Institute, Hefei, Anhui, China*
*youjuan007@sina.com, ayou144@163.com*

**Keywords:**Formal Methods, Software Development, Formal Specification, Formal Verification

## Abstract

Formal methods have been well used in software development segments of requirement analysis, system design and realization, test and maintenance.In this paper, the research status, achievement and problems of formal methods are summarized. Some recent progress and applications of formal methods using in software development are reviewed. The limitations and further research directions are pointed. Research on formal methods can ensure the software system security and reliability by making strict and precise specifications, and verifying several system performances as well.

## 1 Introduction

With the software system complexity increasing, how to develop correct and reliable software has been a continual problem to be settled urgently. An effective way to solve it is the formalized theory, which is an instrument for both software and hardware specifications and system confirmation on the base of rigorous mathematics. By the use of unitary and normative mathematical symbols to design the whole state space, formal methods make a comprehensive frame and attribute structure in depicting, developing and verifying system.

To guarantee the system security, reliability and correctness, formal methods can be used in the whole software development segments that are needs analysis, system design and realization, system testing and maintain. Formalized software development is a process by inferring and verifying to judge whether the product can satisfy software needs and features described in formal or not. It has been of momentous current significance to research on formal methods with the application and progress of software development.

The research on formal methods dates back to 1950's as the period of programming language and compilation technology developed form handicraft style to a systemic method with reliable theoretical principle. The tide of research on formal methods begins with the later period of 1960's. In order to solve the "software crisis" at that time, researchers began to promote several measures with a view to formal methods. They proved whether the computer programming was free form design and coding defects by mathematics. As the definition of formal methods, rigorous mathematics is used because it has the following advantages. Firstly, mathematics is an accurate model medium which can describe object and operation in brief and precise. Secondly, mathematics can represent abstract unity and coherence that are the essence of system specification. Thirdly, mathematics offers a way to show the contradiction and imperfection in specification and the coherence level between design and specification. Formal methods have been used in both software and hardware domain widely and have gotten remarkable result until the later period of 1980's. From then on a new upsurge of formal methods has been set off. Some formal theory, methods and technology have matured gradually in both academic circles and engineering circles, most of which have been used in wide.

The rest of this paper is organized as follows. Review recent progress and applications of formal methods using in software development. Then point some limitations in formal theory and methods. Finally, prospect the further research direction of formal method in software development.

## 2 Classification of Formal Methods

Formal methods can be classified in different way according to different standard.

(a)Formal methods are classified as model-oriented and attribute –oriented king according to target system illustration style.

1)By model-oriented formal methods we mean illustrate system behaviors by constructive mathematical model.

2)By attribute-oriented formal methods we mean define system behaviors indirectly by describing software system attributes.

(b)Formal methods are classified as the following five kinds according to the expression ability [1].

1)The method based on modeling which makes up system model by defining states and operations definitely, such as Z language, VDM[2] and B language. The method represents non-functional needs and time need, etc. though it is not fit for concurrency.

2)The method based on logic which describes expected system propertied including bottom specification, time sequence and possible behaviors by constructing and expanding logic with concrete programming. Then explore system through refining operation set correctly and prove expectation by axiom system related to the chosen logic. Such as ITL (Interval Time Logic), DC (Duration Calculus), Hoare Logic, modal logic, time

sequence logic, TAM (Time Agency Model), RTTL (Real-time Time Logic).

3)By algebraic approach we mean make an explicit associative definition of different operation behaviors in undefined state, such as OBJ and Larch algebraic specification language. It does not give concurrent explicit expression the same as model method.

4)By process algebraic method which allows concurrent explicit expression we mean express system behaviors by restricting the communication among all permitted observable processes. Such as CSP (Communication Sequence Process), CCS (Communication Calculation System), ACP (Algebra Communication Process), LOTOS Language, TCSP (Timing Communication Sequence Process) and TPCCS (Timing Probability Calculation Communication System).

5)The method based on net which is seen as a common expression by graph. It is easy to understand and use for some laypeople, such as Petri, Timing Petri and state diagram.

Some of the above formal methods which almost have similarities on the basis of set theory and predicate logic in technology are combined together to use. Distinctions of them are not definite during formal methods application. By formal language, concrete formal methods and techniques we don't talk about so much in the paper.

## 3 Application of formal methods in software development Classification of Formal Methods

Nico Plat [3] took an example of the application of formal method VDM in DoD-STD-2167A model to answer the question how could formal methods be used in software development. He also made b comparison between formal methods and informal methods to show the shortage of informal methods and advantage of formal ones. Bowen [4] put forward ten practical proposals for formal methods utility. Craigen[5] expounded his views of formal applications in large-scale software development programming. He took several examples in multiple domains, in which developers, development requirements, development contents of item, service tools and formal methods utilities in the whole development course had been explained in detail. Moreover, Jiantiao Zhou [6] studied formal verification technology in workflow model. Generation test cased by making a definition of automatic state transferred path with formal description, Shuai Wang [7] solved the state explosion problem in software development automatic integrated test.

There are mainly two apply styles generalized from formal methods, that are formal specification and formal verification. Formal specification is the description of program function with accurate semantic analyses in formal, which includes mathematical notations, formal specification languages and corresponding tools. Formal specification is the start point of program design and composition, as well as basis for verifying programs. Formal verification concludes model checking, theorem proving and other proof and verified methods. The relation between formal specification and formal verification can be shown as figure 1.

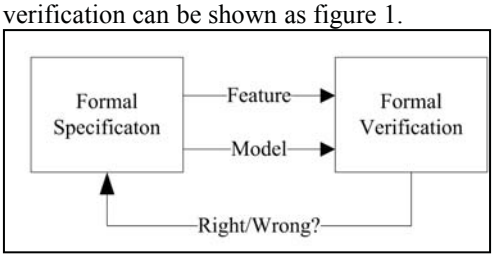


**Figure 1.** Application style of formal methods.

### 3.1 Formal Specification

In the process of software development, we hope to provide some analysis methods and tools to check the feasibility and security of system needs demonstration, manufacture and evaluation before operation. Formal specification overcomes shortcoming of ambiguity by traditional informal methods and has strict formal grammar and semantics based on mathematic logic description which can precisely define coherence, completeness, correctness, specification and implementation. The system which has been deduced and confirmed as correct and safe by automatic or half-automatic formal tools can be concretely carried out and exampled.

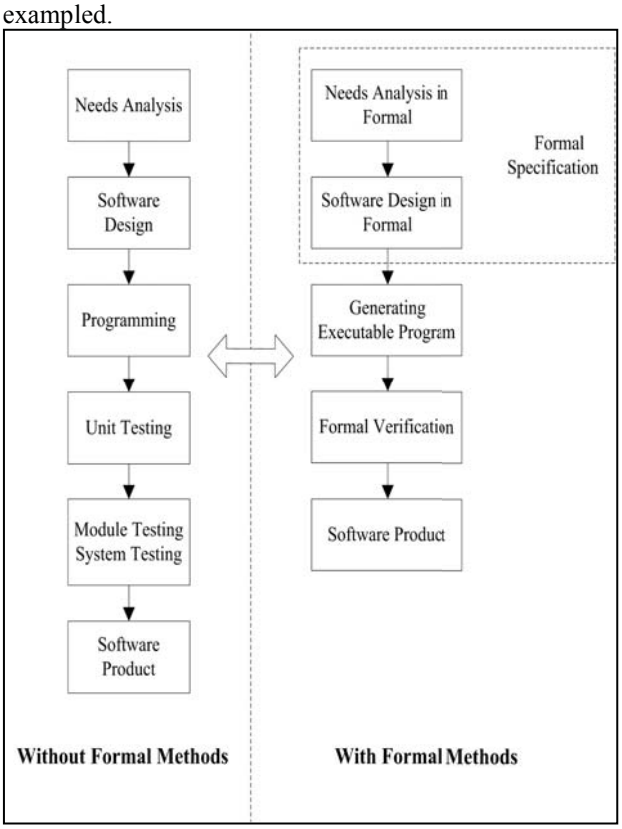


**Figure 2.** Comparison of software development with and without formal methods.

Form figure 2, we can get a clear comparison between the applications in software development with and without formal methods.

First of all, as the base of software development, needs analysis in informal description easily cause inconsistency,

mistaken design and program which would pay a large price to alter after operation. And programs won't be used reliably for the undiscovered mistake. Software design in informal description has ambiguous semantic and more experiences have be depended by participators in design. While formal methods can describe specification from system high-structure in abstract, infer and check it by automatic or semi-automatic tools as well.

An integrated software system formal specification concludes constant, variation and the related operations which are derived from various static characteristics and dynamic behaviors during design and realization. With the adoption of formal methods, the correctness of system strategy execution in theory is proved obviously; the mistake in working is lessened greatly and the system secure reliability is increased.

### 3.2 Formal Verification

In accordance with the above formal specification, formal verification is the mathematical analysis course of performance proof and evaluation based on established model and close reasoning. Formal verification proves system properties dispensed with operation [8]. It can demonstrate the correctness and correspondence between specification and process in formal test and verify. That is to say system software and hardware development have the same determinacy as mathematic proof.

Formal methods were used in verifying high reliability software at first-phase of software development period including of analysis and design. Formal verification can test hidden perils and errors to maximum, thus dispelling some system design defects. The development of formal verification is shown as: BAN logic period → several trusted logic based on BAN logic period →mode checking period→theorem proving period [9].

The basic methods of formal verification conclude model checking [10] and theorem proving [11].

(a)Model Checking

Model checking is a finite state system automatic verification technique to check whether system performance can satisfy expected qualities by traversing all finite state room in system model. The merits of model checking are the high automation and fixing fault position by generation counterexample as soon as some character does not satisfy. The main demerit of it is the problem of state room explosion. There are two methods in solving the problem. One method depends on the structure traits of model state room. The main concludes symbolic model checking, symmetric model checking, partial ordering model checking and on-the-fly model checking. The other method decomposed complicated system verified into disposal parts in the abstract. The main concludes abstract methods, composed methods and so on.

(b)Theorem Proving

Firstly form a formal system on the base of axiom and inference rules after model in logic for software system and related characters. Then give full proof of axiom to verify whether the system possesses expected features or not. The advantage of theorem proving is it can solve infinite state space by induction in mathematic infinite region. The disadvantage of it is less in automation and cannot give a counterexample when formal verification fails.

The formal verification on software function and security can be well realized by intelligent use of above methods.

## 4 Limits of formal methods

There are some self-limits in theory and methods for formal even though lots of progress has been achieved in theory research and application. The main limit in theory is concerned about ambiguity while manageability caused by lack of effective operation tool is prime problem in actual application. The applicability of formal methods used in developing larger software system and the scope of application in solving special problems are still ambiguous [13].

Firstly, formal methods can reduce the risk of specification mistake and misunderstanding but eliminate it. It is necessary to treat formal specification reasonable on ambiguity and accuracy when system formal specification and formal verification can increase the possibility of plan correctness but absolutely right. Because of some inaccurate and in complete environment description or human factors in formalized course, system formal specification can bring some false implementation. Even if it has a right system operation performance, it still will get a wrong plan with a wrong formal specification.

Secondly, a basis problem of formal methods is it hard to specify some non-functional needs and characters accurately at present. Formal methods cannot be used in all details. It has great expression ability in constitutive property but low in functional expression. That is to say, we still need combine informal methods with formal methods to describe non-functional features.

Thirdly, formal methods are "choice" [14]. By use of formal methods, system accuracy and reliability increase. So system only pays more attentions on accuracy and reliability can suit for formal description. It does not have a significance to use formal methods for the system concerned about interface figure or software operation rate. The system in less use of algorithm whose complexity mainly embodies plenty of input and output is not suitable for formal specification either. For time expenses and redundancy of formalization will be the same as cost of real system cost in case of formalization. It becomes unacceptable to verify in formal every time as soon as some changes.

Furthermore, because system performance mode is described by mathematics in formal, there are still limits of formal methods in theoretical and technical system training for system designers, testers and evaluators. Strict description, inference and verification become necessary for system especially for large scale one which will increase expenses.

Finally, lack of effective formal verification tools is desiderated to solve and new formal theory and methods need to be further research as well.

## 5 Research prospect

Make further approach and research on formal methods has been a long-time and arduous task. Analyze and describe all of system characters effectively is hard for formal methods even though each formalization has its own advantage. It is necessary to further research and perfect formal theory and technique on the base of existing software development methods. Enhance the analysis simulated and verified ability to make better use of every stage of software development. We make some prospects for formal methods further research.

Firstly, standardize formal methods and techniques. Only with more uniform standardization can improve the description ability and application scope of both formal specification grammar and semantics.

Secondly, one of critical factors restricted formal method development all along are the reliability of formalized tools. How to measure and improve tools' quality and ability has been a long-term research hot spot.

Thirdly, formal verification technology has achieved multiple achievements based on theoretical research and engineer practice. With the quality need improvement and software product and system complexity development, more requirements in concurrency, no determinacy, distributed control and secure reliability have been emerged. So how to perfect and improve the ability of existing techniques to solve problem and how to develop new method to adapt software development requirements have been the priority in formal methods research.

Fourthly, at the test stage of software system development, formal methods can be wide used in test suit generation because of the characteristics of generating counterexamples in formal verification. Especially during the test stage, formal methods provide logical abstract description methods for system developer and evaluator, which have been an effective complement for cognizing behaviors of code-test system. Formal methods can also describe and infer simulated for system behaviors and generate test suits, choose test data and evaluate the test coverage al wee. Son on the ground of formal methods, we can have a comprehensive, reliable and active reorganization on objective system conducts [15]. At present, a definite theory explanation combined formal verification with test methods has not been published in public both at home and abroad. It still needs to further research and explore. Moreover, how to combine existing formal verification theory with consistence test theory to form new one, how to define the frame and procedure of formal methods, how to standard the unified semantic description and how to improve related techniques of formal inference and test all have been crucial problems to deeply analyze.

## 6 Conclusion

In a word, to recognize the position and affection of formal methods in software development in reason and to guide and help we inquire and solve problems engendered during software development are the essential aims of formal research. Make clear the application of formal methods will exploit more internal advantages and latent capacity to every stage of software development.

## Acknowledgments

I'd like to express my sincere thanks to all those who have given me encouragement, support in the course of my writing this paper.

## References

[1] Barroca L M, et al 1992 Formal Methods: Use and Relevance for the Development of Safety-Critical Systems[J] . The Computer Journal. 35(6).

[2] Guttag J and Horning J 1993 Larch: Language and tools for formal specification R[ J]. Springer-Verlag.

[3] Nico Plat, Jan van Katwijk and Hans Toetenel 1992 Application and benefits of formal methods in software development[J]. Software Engineering Journal. Sep.7(5):335-346.

[4] Bowen J P, et al 1995 Ten Commandments of Formal Methods [J]. IEEE Computer. Apr,28(4):56-63.

[5] D Craigen, et al 1995 Formal Methods Reality Check: Industrial Usage [J]. IEEE Trans. Soft. Engi.35(5).

[6] Zhou Jian-tao, Shi Mei-lin, Ye Xin-ming 2005 Formal Verification Techniques in Workflow Process Modeling [J]. Journal of Computer Research and Development.42(1):1-9.

[7] Shuai Wang, Yindong Ji, Wei Dong, Shiyuan Yang 2010 A new formal test method for networked software integration testing [A]. In: Proc. Of ICCSA 2010 [C].LNCS 6017, BSV2010:463-474.

[8] Devayne E Perry and Alexander L Wolf 1992 Foundation for the Study of Software Architecture[A]. ACM SIGSOFT Software Engineering Notes[C]. 17(4):40-52.

[9] Jan Wessels 2001 Applications of BAN-LOGIC[C]. IPA Spring Days on Security. April 18-20.

[10] Clarke E M, Orna Grumberg and Doron Peled 2000 Model Checking[M]. Cambridge: MITP.

[11] Clarke E M, Wing J M 1996 Formal methods: State of the art and future directions[J]. ACM Computing Surveys. 28(4):1-22.

[12] Godefroid P, Huth M 2005 Model checking vs. generalized model checking: semantic minimizations for temporal logics[A]. In: Proc. Of 20th Annual IEEE Symposium on Logic in Computer Sciences[C]. Chicago, USA:IEEE CS Press:158-167.

[13] Ehrig H 1995 Theory and Practice of Software Development[J]. EATCS, No 57.

[14] Ralf Kneuper 1997 Limits of Formal Methods[J]. Formal Aspects of Computing.9:379-394.

[15] Ye Xin-ming 2009 A Survey on the Methodology Integrating Formal Verification and Conformance Testing[J]. Journal of Inner Mongolia University. 40(4):486-492.