UDC 681.3.068

A.V. GAHOV, V.O. MISHCHENKO

V.N. Karazin Kharkov National University, Ukraine

THE VALIDATION OF THE SOFTWARE THAT WAS DEVELOPED FOR CALCULATIONS RELATED TO THE DESIGN OF ANTENNAS

We consider validation problem for the software of calculations related to the design of antennas under critical requirements. According existing standards, we propose the quality model for this software. The model is based on the system of standard (mainly) and special metrics. Additionally, concerning the software reliability, we suggest that it will be very important to check the compliance of development of software with its generalized description. To verify whether our method is lively, we consider two samples of real programs and assess relating internal or external metrics and the compliance characteristic of one of development processes.

validation, software, quality, diffraction, antennas, internal and external metrics, measurement, process

The original problem and the challenge

Modern telecommunication (in particular, mobile) systems make new very critical demands to used antenna devices. The main conditions are that they must be of very broadband (multiband) frequency response and of compact size (e.g. [1]). To find the appropriate form of the antenna designer can use computer modeling of spatial processes of interaction of electromagnetic waves with opened metal screens (e.g. [2,3]). For concreteness, we will mean scattering waves on perfectly thin perfectly conducting screens. We limited ourselves to the modeling, based on the hypersingular integral equations (HSIE) for the appropriate density functions (e.g. density of surface currents).

The program, which implements the calculation of this type, is EDEM3D [3]. Reliability of calculations with the help of this program is based on a mathematical theory; and it is confirmed by reports about successful applications [4]. However, the information on the evaluation of the quality of the product in accordance with a recognized model is not published. This makes it difficult for objectively compare EDEM3D with the same software. At the same time, modern standards allow to build models of software quality for all applications [5].

Note that more complex problems with scattering waves have been solved lately [6], and related algorithms and software are being created now[7]. Consequently, the building of the software quality model is important and urgent.

Different aspects of the quality of such programs were regarded in [8-10]. In present work, the problem of the validation of programs of modeling the spatial scattering examined based on international standards ISO/IEC 9126 and IEEE 982 [5].

Note that in these applications the reliability of the calculations can depend on adequacy of mathematical model, and on accuracy of discretization of this model, and on effectiveness of computational scheme, and on reliability of coding-testing process. In connection with the problem complexity, verification (validation) of such products is extremely complicated.

Problem statement

Our first task was to build a system of metrics that could meaningfully compare different software for mathematical models of 3D diffraction on unclosed metal screens (or multiple screens). The system of metrics should reflect, as far as possible, the relationship between the characteristics of software quality and balance them with the processes of the life cycle.

It should also be borne in mind that the developer is one researcher or a very small team. That is, the developer must be able to calculate the metric internal quality himself, with the lowest cost.

Calculation similar metrics based on the external image of the software should also be available for the simplest expertise.

The second task was to check the feasibility of calculating selected metrics, particularly special. We assessed also time spent on assessing value of external metrics.

The software, which under consideration has a following feature. The usefulness of this software depends primarily on the reliability of the calculation fields in the specified points regardless of the remoteness of these points from the screen. However, it is very difficult to obtain accurate experimental data for many cases. There is also no suitable analytical expression for the scattered fields for any cases. Consequently, the tests of external quality cannot be based on the direct comparison of calculated and real fields. You can only check the consistency of results in a series of calculations that is probable in the case for the correct calculations and impossible in the case of incorrect ones.

Similarly, the analytical verification of the code is not feasible because of the complexity calculations. Therefore, in the internal testing we had to rely on a few signs. We need to build confidence in the reliability of product through evaluation of the software process.

We used this idea under common recommendations of the IEEE 982 standard. In our case we had to find a method that would have been attractive because of its cost-effectiveness and ease of interpretation.

Proposed method and software quality model

To calculate the metrics that describe the quality characteristics, it is necessary to define the functions provided software. For the software, which uses the approximation HSIE to calculate scattered fields, the list of these functions is as follows:

IO_1 – support of the input of geometrical parameters;

IO_2 – choice of the fallen field, and other physical characteristics;

IO_3 – control of the parameters of discrete model;

BC_1 – building of basic equations;

IO_4 – saving of basic equations and their restoration;

BC_2 – solving of the basic equations;

IO_5 – saving the solution and its restoration;

AR_1 – request and demonstration of the basic equations and solution of them;

BC_3 – calculation of fields at certain points;

BC_4 – calculation of the directional diagrams;

BC_5 – calculation of currents (of vector fields);

AR_2 – output of numeric modeling of the fields at certain points;

AR_3 – visualization of the results of the modeling;

FI_1 – request and output check information;

AR_4 – request and output of the directional diagrams;

AR_5 – request and output of the calculated currents;

AR_6 – request, calculation and output of integral scattering coefficients;

BC_6 – extra calculations based on the equations (BC_1) or the solution (BC_2);

FI_2 – informing user about current events, and providing irregular control (e.g. ‘stop’);

FI_3 – informing user, as in help process;

FI_4 – supply of user interface as a whole.

Developers have to complete their lists by special functions, if any. It should also divided one function for a few sub-functions, if this function’s sense is overburden. Our functions affected in next tasks:

– Input-output of data needed for calculations (IO);

– Base calculations (BC);

– Analysis of results (AR);

– Friendly interface (FI).

It is allow changing the distribution of the functions over the tasks. The designer must specify goals for the tasks and specifications for the functions.

Following ISO/IEC 9126.3(2) standard, we have chosen the following metrics of internal (external) quality. Our special metric marked with ‘spec’.

1. Functionality

1.Suitability metrics (0.5): 1.Functional adequacy, 2.Functional implementation completeness, 3.Functional implementation coverage.

2.Acuracy metrics (0.2): 1.Computational accuracy, 2.Precision (if any data element requires defined level of precision).

3.Interoperability metrics (0.1): 1.Data exchangeability – Data format based.

(4.Security metrics are omitted because of they depend on terminal application).

5.Functionality Compliance metrics. (0.2): 1.Functional compliance.

2.Reliability

1.Maturity metrics (0.5): 1.Fault removal (Estimated latent fault density), Test adequacy (Test coverage).

2.Fault Tolerance metrics (0.333): Incorrect operation avoidance.

3.Recoverability metrics (0.333): spec – Worst data restoration (spec - Worst effectiveness of restart).

4. Reliability Compliance metrics (0.333): Reliability compliance.

3.Useability

3.1.Understand ability metrics (0.4): Completeness of description, Demonstration capability (Demonstration accessibility), Function understandability (Understandable input and output).

2.Learnability metrics (0.2): Completeness of user documentation and/or help facility (Help accessibility).

3.Operability metrics (0.2): Input validity checking (Error correction), Customizability, Message Clarity (Message understandability in use), Interface element clarity (spec – How march interface informative).

4.Attractiveness metrics (0.2): Attractive interaction (Attractive interface), Interface appearance customizability.

(5.Usability Compliance omitted because of unknown use context).

4.Efficiency

1.Time Behavior. Metrics (0.5): spec – Asymptotic power for turnaround time (spec - Worst case of time power ratio).

2.Resource Utilization metrics (0.5): spec – Asymptotic power for utilized memory (spec – Worst case of memory power ratio).

(3.Efficiency Compliance optional because of unknown conditions of use).

5.Maintanability

1.Analyzability metrics (0.2): 1.Readiness of Diagnostic function (Diagnostic function support), 2.Activity recording (Audit trail capability).

2.Changability metrics (0.4): 1.spec – Units of classes and templates (Parameterized modifiability), 2.Change recordability (Software change control capeability).

3.Stability metrics (0.2): Modification Impact localization.

4.Testability metrics (0.2): Completeness of built-in test function (Re-test efficiency), Test progress observability (Availability of built-in test function)

(5.Maintainability Compliance is omitted because of unknown requirements).

6.Portability

6.1.Adaptability metrics (0.6): Adaptability of data structures, System software environmental adaptability, Porting user friendliness.

2.Installability metrics (0.2): Installation effort (Ease of installation).

3.Co-existence metrics (0.2): Available co-existence

(4.Replaceability, 5.Portability Compliance are optional because of undefined use conditions).

Now, as announced above, we determine an additional metric of software process. To calculate relating attributes you have to present the code development as process, which producing sequential versions. Then you must investigate specifications of each version to prescribe them general descriptions, such as:

– improvement of structure without essential change of a code;

– search of defects and make local correction of a code;

– essential completion of functions according to recent reorganization of the interface and structure,
and so on.

The "Energy compliance" metric rely on the value of "intellectual heat" [11], that is

$$Q = E - A, \quad (1)$$

where E – specification energy, A – work of programming of same version (Holstead's scientific metrics are the origins of these attributes" [11]). There are four variants:

(M++) – all descriptions of the versions comply to the Q signs (and values) relative to the same versions;

(M+-) – compliance is incomplete, but it have a simple explanations;

(M-+) – compliance is not quite full and have not a simple explanations;

(M--) – no any compliance.

Test of calculations for real examples

We tested the applicability and the ease of metrics calculation on two examples. Firstly, we verify the profile of internal quality for the Diffraction_On_Rectangle application [12]. Then we tried to determine profile of external quality for the demo of EDEM3D program [3]. Let us preliminary specify sense of our special metrics.

"*Worst effectiveness of restart*" metric is **external** one. It purpose is "How effective is the restart capability?" The method of application is "Count the time saved after program restoration". Measurement formula is X=1–A/B, where A – expand time for restart calculation from the halt point in the worst case, B – time from the start up to the halt (0<X<1).

"*Worst data restoration*" metric is **internal** metric. It purpose is "How effective is the restoration capability?" The method of application is "Count the part of the data which can be restored after program halt". Measurement value is X=A, where A – max part of data which can be restored in the worst case (0<X<1).

"*How much interface informative*" metric is **external** one. It purpose is "What part of elements can user understand from the interface?" The method is "Evaluate the part of interface elements that is fully understand to user in context of current situations". Measurement formula is $X = A / UOT$, A – total number of interface elements (0<X<1).

"*Asymptotic power for turnaround time*" metric is **internal**. It purpose is "What is the power in asymptotic formula for time to complete a group of related tasks as a job lot?". Measurement formula is X=a, where $cN^a \approx$ time of main loop, N – parameter of discretization (1<X<∞).

"*Worst case of time power ratio*" metric is **external**. It purpose is "What is the worst ratio of theoretical and estimated powers for time asymptotic formula?". The method is "Estimate the time T to complete a group of tasks as a job lot. Assess such power b that $T \approx cN^b$ and compare with power attribute 'a' – from the metrics of "Asymptotic power for turnaround time". Measurement formula is X=max{min(a, b)/max(a, b)} (0<X<1). For calculate the parameter b expert may use regression.

"*Asymptotic power for utilized memory*" metrics is **internal**. It purpose is "What is the power in asymptotic formula for memory size that the product (in main loop) will occupy to complete a specified task?" Measurement formula is X=a, where $cN^a \approx$ the memory size, N – parameter of discretization (1<X<∞).

"*Worst case of memory power ratio*" metric is **external**. It purposes is "What is the worst ratio of theoretical and estimated powers for memory asymptotic formula?" The method is "Estimate the size M that the product (in main loop) will occupy to complete a specified task. Assess such power b that $M \approx cN^b$ and compare with power attribute 'a' from the metrics of "Asymptotic power for utilized memory". Measurement formula is X=max{min(a,b)/max(a,b)} (0<X<1). For calculate the b expert may use regression.

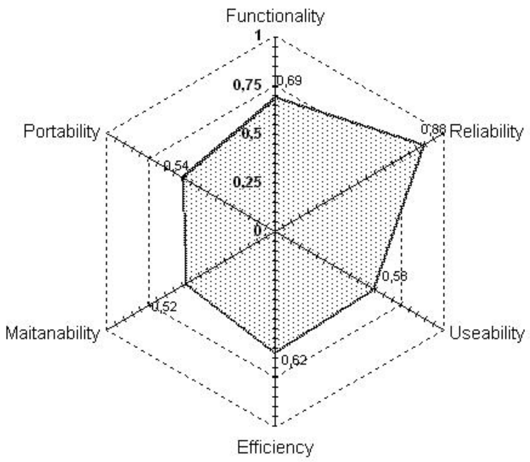
"*Units of classes and templates*" metric is **internal**. It shows which part of software units concerned with classes, templates or their analogues.

For the Diffraction_On_Rectangle all required internal metrics have been calculated (Table 1, Diagr. 1).

Table 1

Fragment of the report of evaluation internal metrics

Metric Number	Value and Range (or Scale Type)	Reduced Value in 0.0..1.0
1.1.1	0,86 in 0.0..1.0	0,86
1.1.2	0,66 in 0.0..1.0	0,66
1.1.3	0,76 in 0.0..1.0	0,76
1.2.1	0,60 in 0.0..1.0	0,60
1.2.2	0,60 in 0.0..1.0	0,60
1.3.1	1,0 in 0.0..1.0	1,00
1.5.1	0,46 in 0.0..1.0	0,46
2.1.1	0,75 in 0.0..1.0	0,75
2.1.2	0,38 in 0.0.. ∞	0,28
2.2.1	0,90 in 0.0..1.0	0,90



Diagr. 1. The internal quality diagram of the Diffraction_On_Rectangle

Table 2

General descriptions compliance to values Q of "intellectual heat" on example of software process Diffraction_On_Rectangle [13]

Version	**General description**	***E***	***A***	***Q***
19-March-2003	Elementary realization of simple solution	8,98	4,.64	4,3
12-May-2003	Structure simplification	6,.39	7,.63	–1,2
16-Nov-2003	Cleaning and preparation for expansion of functionality	12,53	6,62	5,9
19-Jan-2004	Modification: expansion of functionality	5,89	8,59	–2,7

Check of an "Energy compliance" metric for the Diffraction_On_Rectangle has yielded the results presented by Table 2. Assessment has accorded to M++:

Concerning some results of the measuring of the metrics of external quality of our model see Table 3.

Table 3

Time and success for evaluation external metrics of quality of EDEM3D in the demo version [14] (fragment)

Metric Number	Time spent on assessing value (hours)	Reliability of measuring
1.1.1	10,6±3,9	Satisfactory
1.1.2	6,3±4,2	Satisfactory
1.1.3	6,3±4,2	Satisfactory
1.2.1	3,5±6,0	Not reliably (lack of data)
1.2.2	0,0	Not evaluated (no data)
1.3.1	0,.0	Not evaluated (no data)
1.5.1	0,75±0,25	Not reliably (lack of data)
2.1.1	1,5±0,5	Not reliably (lack of data)
2.1.2	0,75±0,25	Satisfactory
2.2.1	1,25±0,25	Satisfactory

Conclusion

We have to say about presented work that for the first time a quality model was build for the software of modeling spatial diffraction on screens. This one was based on modern international standards ISO/IEC 9126 and IEEE 982.1. In this model, reliability, as an important aspect of quality, has been reinforced by the metric of software development.

Testing on real examples confirms the viability of a model in practice. Developers of the Diffraction_On_Rectangle product checked feasibility of measurement for all attributes, which related to internal metrics of the model. On the other hand, authors had measured most of those attributes of EDEM3D product [14], which are prescribed by external metrics (excluding few attributes which have not be meaningful for the demo version).

We think that the applied value of this work is primarily in the next. Firstly, the diagram of internal quality of the well-established Diffraction_On_Rectangle product can orientate developers of similar products in desirable levels for each of the six characteristics of the software quality. The second benefit stems from our assessment complexity of validation of external quality of an example of the software. We have estimated the labor for such validation. The estimation was 47±16 man-hours on example of the demo version of EDEM3D.

The experience which presented by our work will be useful and in more general situations. These are such cases where it is need to answer the question about the software based on nontrivial mathematical theory: Do this software suitable for dependable calculations?

References

1. Nathan Cohen Fractals' New Era in Military Antenna Design // RF Design Defense Electronics Supplement, August 2006 issue. – [Электрон. ресурс]. – Режим доступа: (http://rfdesign.com/issue_20050801).

2. Jirí Horák, Petr Chmela, Lukás Oliva, Zbynek Raida. Multiband planar antennas on electromagnetic bandgap substrates: Complex global optimization of the structure // Microwave and Optical Technology Letters. – 2006. – Vol. 48, Issue 12. – P. 2532 – 2534.

3. Davydov A.G., Zakharov E.V. Opportunities of program EDEM for development of devices of antenna techniques // Antennas. – 2006. – Issue 10. – P. 52-57.

4. Davydov A.G., Zakharov E.V., Pimenov Yu.V. Numerical Analysis of Fields in the Case of Electromagnetic Excitation of Unclosed Surfaces // Journal of Communications Technology and Electronics. – 2000. – Vol.45, Suppl.2. – P. S247-S259.

5. Kharchenko V.S., Sklyar V.V., Tarasyuk O.M. Methods of modeling and estimation of quality and reliability of the software. Manual – Kharkov: National aerospace university “KhAI”. – 2004. – 159 p. [in Russian].

6. Gandel Yu.V., Mishchenko V.O. Pseudo-differential equations of electromagnetic diffraction on a plane-parallel structure and their discrete model // Bulletin of V. Karazin Kharkiv National University. – 2006. – №733. Series “Mathematical Modelling. Information Technology. Automated Control Systems”. – Issue 6. – P. 58-75 [in Russian].

7. Gahov A., Mishchenko V. The Hypersingular Equation Technique for the 3-d Problem of Diffraction on a Metal Screen in a Stratified Medium // Proceedings of the 11th International Conference on Mathematical Methods in Electromagnetic Theory. – Kharkiv, Ukraine, 26-29 June 2006. – P. 602-604.

8. Anfinogenov A.Yu. Increase of computing efficiency of the solving of problems of the theory of diffraction by a method of discrete features // Proceedings of XII International symposium "DSMMPh". – Kharkov-Kherson. – Lazurnoe, 13-18 June 2005. – P. 13-16.

9. Mishchenko V.O. Software of DSM: a role of mathematical models of reliability and labour input // Proceedings of International school-seminars «DSMMPh». – Orel. – OSU. – 2006. – Vol. 4.– P.73-80 [in Russian].

10. Gahov A. Efficiency of calculations by DSM in the 3d scalar diffraction // Proceedings of International school-seminars «DSMMPh». – Orel. – OSU. – 2006. – Vol. 4. – C. 23-29 [in Russian].

11. Mishchenko V.O. Mathematical model of style Software Science for the metric analysis of complex scientific programs // Bulletin of V. Karazin Kharkiv National University. – 2004. – №629. Series “Mathematical Modelling. Information Technology. Automated Control Systems”. – Issue 3. – P. 70-85 [in Russian].

12. Gahov A.V., Mishchenko V.O. Computing experiment on the basis of the numerical decision hypersingular integral equation for rectangular area // Bulletin of V. Karazin Kharkiv National University. – 2003. – №590. Series “Mathematical Modelling. Information Technology. Automated Control Systems”. – Issue 1. – P. 84-91 [in Russian].

13. Gahov A. Testing a new approach to the analysis of projects development using generalization parameters offered by software science // Transactions of International Conference SCALNET’04, 28-30 September 2004. – P. 118-120.

14. Manual and Demo of EDEM3D [Электрон. ресурс]. – Режим доступа: http://edem3d.ru/.

Поступила в редакцию 7.03.2007

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.