16

December 2011

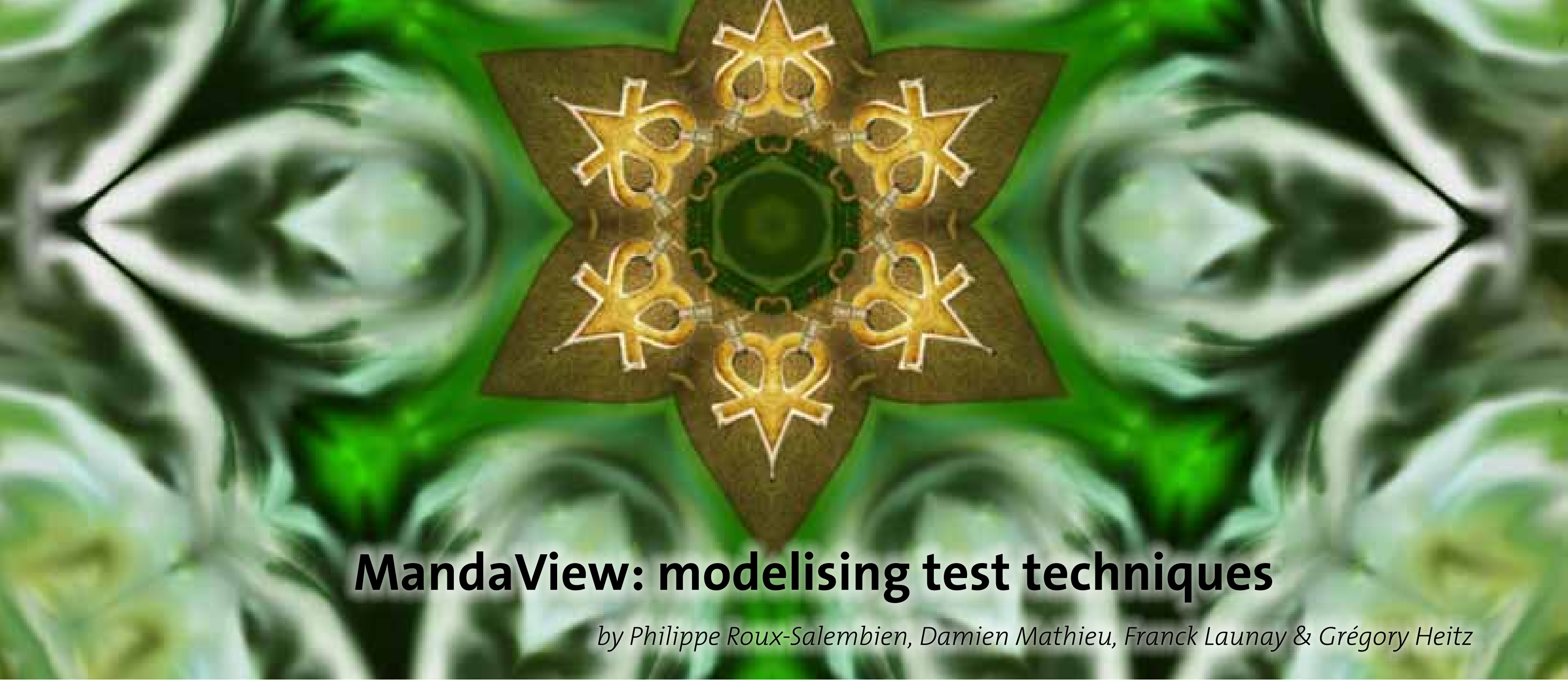
testing experience

The Magazine for Professional Testers

The Future of Testing

© iStockphoto.com/pidjoe

ISSN 1866-5705 www.testingexperience.com free digital version print version 8,00 € printed in Germany



MandaView: modelising test techniques

by Philippe Roux-Salembien, Damien Mathieu, Franck Launay & Grégory Heitz

We have all attended training courses, presentations and seminars, and have seen a wide variety of schematics and graphics used to illustrate ideas and concepts. Do we really remember them and do they really help us to learn and understand?

From a pedagogical point of view, this is a real concern. Illustrations are far more effective than lengthy texts to emphasize ideas, but they can become less effective when there are too many.

With this in mind, we have devised a new concept: MandaViews (MDVs). MDVs provide a family of models for demonstrating test techniques. The goals are on the one hand to provide pedagogical ideograms to facilitate the understanding and long term memorizing of all of the different categories of test techniques (black-box, defect and experience-based and white box), and on the other hand to provide a meaningful project reporting tool for managers.

Why suggest a model for test techniques modelization?

A pedagogic need

Many of the different techniques are not only well understood but also memorized and organized by testers who want to improve their skills. To achieve a good and effective understanding can involve a considerable amount of work: compiling paperwork, creating checklists etc.…

A documentation approach

Furthermore, providing an overview of all the different test techniques actually used in each step or level of a testing project could provide valuable feedback to the stakeholders on the range and the complementarity of the panel of test techniques currently in use.

Mandalas: from ancient tradition to applied pedagogy

In Sanskrit, mandala means circle, and these concentric diagrams are commonly used as a meditation tool.

For several decades now, this ancient concept has been adapted and used by teachers to illustrate related concepts and notions, making them more understandable[1].

Philippe Roux-Salembien attended a conference on ‘How can we increase our effectiveness in helping our children do their homework’, held by senior teacher Armelle Géninet at his daughter’s secondary school. Mme Géninet explained, among different concepts and ideas, how the philosopher and educator Antoine de la Garanderie had developed the ‘Mental Management’ idea and its 5 key elements to describe the entire learning process: Attention, Memorization, Comprehension, Reflexion and Imagination[2]. This research identified the need for the learner to ‘spatialize knowledge’. The mandalas described above are used to assist in assimilating knowledge in fields as diverse as mathematics and English. The specific use of mandalas in such pedagogic applications compared to other graphic models, relies on the use of symmetries, graphical links and codes between the different items. The combination of these enhances spatialization, understanding and memorization of the concept to be learned.

During this conference, Philippe had the idea to apply those pedagogic mandalas to his own particular field, of testing, and more specifically, use it to represent test techniques. This idea was supported by the variety of available test techniques in each category as described in the ISTQB syllabi. He therefore designed a first model for black box and experience based techniques and named this family of mandalas dedicated to testing: MandaView. Some time later at a meeting with Damien Mathieu, Franck Launay and Grégory Heitz, Philippe discussed this concept with them. They agreed on the originality of the idea and discussed the potential of MDVs in testing-related activities. From there they decided to continue exploring other uses for MDVs that could be shared with the international tester community. This is how the idea of this article came about.

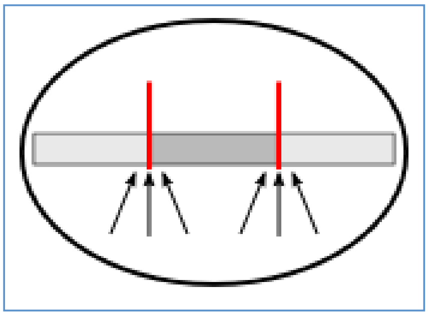
[1] *‘Graphismes et mandalas d’apprentissage’, CP, A. Géninet, éd RETZ*

[2] *‘Les profils pédagogiques’, A. de la Garanderie, éd Bayard*

MandaView basics

Like mandalas, MDVs are based on circular diagrams. These circles are divided into sections to represent each item. Each section contains a graphic symbol (ideogram) illustrating the test technique that it represents.

The first principle of understanding MDVs is the clear association between the ideogram and the principle of the related test technique.



Example: Ideogram for boundary value analysis

Thus, the process of remembering the core principle of each test technique on a long term basis, even those not often used, is made easier.

Positioning and grouping the techniques according to their affinities and categories also helps to identify and understand their potential daily use.

MandaView Model for effective pedagogy

The three aims of the MDV Model rely on the process shown in Fig. 1.

Firstly, spatialization enables a global approach for all the techniques within a category by grouping related techniques. Secondly, visualization is improved by means of color themes and ideograms to help understanding and retaining the principles of the particular techniques concerned and their relationships. Thirdly, the resultant unified models improve the ability to memorize the information long-term, once some time has been spent initially to understand the entire model.

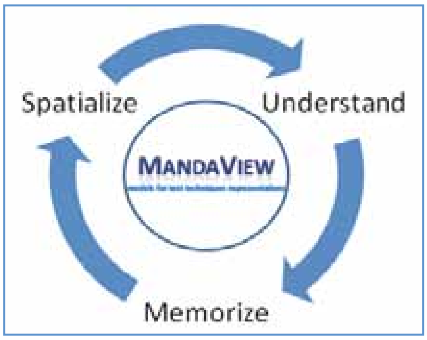


Figure 1: Key features of MDV Model, inspired by Antoine de la Garanderie research

MandaView Model for modelization of specification, defect and experience-based techniques

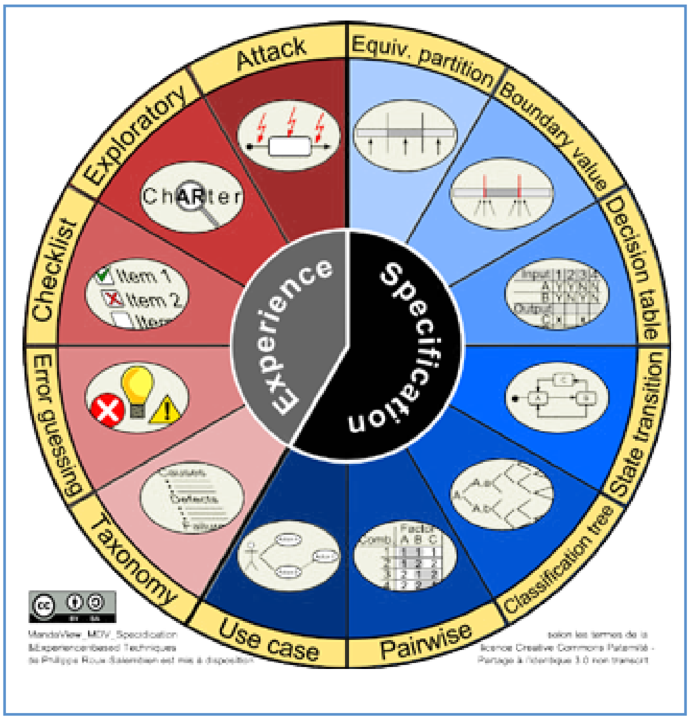


Fig.2: First MDV: example for specification-based techniques and defect and experience-based techniques.

Why grouping black box and experience-based?

Black box and experience-based techniques together probably constitute the test techniques most commonly used by test analysts, therefore it makes sense to group them together in a single diagram. Furthermore, they encompass around a dozen different techniques in accordance with ISTQB, which is ideal for the construction of the first MDV.

The circular shape, which mandala represents and is therefore the foundation of MDVs, enables a quick overview of the test techniques being demonstrated. Specific groupings of related techniques, with associated comments, are thereby possible. Furthermore, dividing the MDV into three areas eases knowledge spatialization.

1. The test techniques' names in the external ring;
2. The ideograms designed to represent the basis of the related technique, in the main ring;
3. The central zone showing the categories within this particular MDV.

The central zone of the MDV shows the limits between the two categories: specification-based on the one hand, and defect and experience-based on the other. These two parts of the picture are also identified by two main background colors: each specification-based technique is in a specific shade of blue; whereas each experience-based technique is in a red shade.

The ideograms relating to the test techniques are easy to insert into the MDVs and provide assistance for understanding and memorization of the core principles of the techniques.

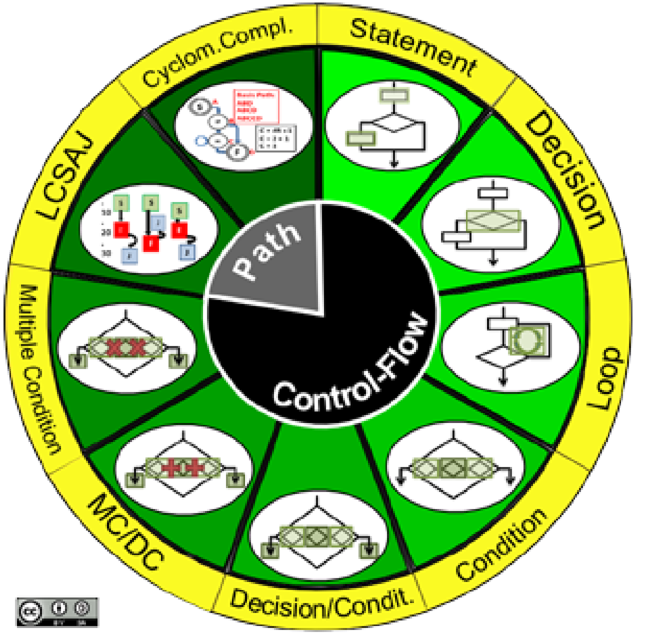
The first two ideograms of specification-based techniques show the key roles of the equivalent classes within them. Then the ideogram of a decision table technique shows an example of a decision table which is most commonly used and easier to identify than a cause-effect graph.

The state-transition ideogram shows the starting point with three different states, each of which relates to the others. The classification tree shows the outline of an example of this tree, whereas pairwise testing is illustrated by showing only a part of the particular table. The use case ideogram displays a simplified view of the usual UML diagrams.

On the defect and experience-based side, we chose to highlight causes, defects and failures in the taxonomy testing ideogram according to the ISTQB definition. Error guessing testing is probably fairly simple to recognize, as is for checklist testing. Exploratory testing shows a magnifying glass zooming on the word "Charter" illustrating the concept of using tester's intuition around predetermined themes. And finally, the software attack ideogram shows lightning strikes, representing the attempts to force failures through interactions with the software and its environment[3,4].

One can observe that the spatialization of the twelve techniques of the two categories enables grouping, which helps both understanding and memorization. Two examples: on the one hand equivalence partitioning testing and boundary value analysis testing are put to one side as they are both based on class equivalence; on the other hand, software attack might use the other experience based techniques which is why it has been put on the last position of these latter techniques.

MandaView Model for modelization of structure-based techniques



MandaView_MDV_White_Box de Philippe Roux-Salembien est mis à disposition selon les termes de la licence Creative Commons Paternité - Partage à l'Identique 3.0 non transcrit

Fig.3: MDV for structure-based techniques

Two main structure-based (white box) techniques: Control Flow Testing and Path Testing

In order to illustrate how useful white box techniques are for testing, we will quote Black & Mitchell[5], whose book was very helpful for writing this chapter (together of course with the above mentioned ISTQB Advanced Level Syllabus): for a 'world class, excellent system test team using all black-box and experience-based techniques', doing all the job the best way, 'at the end of that, have they done enough testing?'… 'May be as much as 70 percent of all of the code in the system might never have been executed once! Not once!'. Please note that all technical definitions used in this chapter are quotes from this book.

We will now focus here on the reasons why we conceived the 7 ideograms for control-flow techniques and the two ideograms for path testing, which were used to constitute the White Box testing MDV. Of course, the structure of the second MDV is similar to the first one.

With the aim to design the different white box techniques ideograms, we chose graphical representations inspired by control-flow graphs, but somehow different in order to concentrate the essentials of each technique in each of them.

As defined in the ISTQB glossary of testing terms, statement testing is a white box test design technique in which test cases are designed to execute some or all of the available statements. The selection of the statements to be tested is based on their being executable. Among all control flow techniques, this technique offers least code coverage, and it is the first ideogram in the White-Box MDV. The black arrow means a set of consecutive statements and the pale green box represents the tests in this part of the code.

The second strongest level of structural coverage is the decision – or branch – coverage. It looks for decision, often 'IF' conditions or 'SWITCH' conditions (which can be seen as a complex set of atomic 'IF'), whereas 'DO…WHILE' or 'WHILE' are loop decisions. As the decision expression is tested in decision coverage, we represent it as a green box, with two emerging arrows related to the TRUE and FALSE outcomes of it.

We included loop coverage in the White-box MDV, despite it not being presented in the ISTQB Advanced Syllabus. The chosen way to represent loop coverage is rather straight: an arrow, covered by statement coverage (pale green box) with aside a loop with a deep green box representing the tests to be designed. Let's note that this technique is very hard to apply extensively, and that the above mentioned Boundary Value Analysis technique is then helpful. This shows rather accurately how black and white box techniques can be and even have to be related.

The fourth structure-based technique is condition coverage. As decision coverage checks if we have exercised each decision both ways, condition coverage discusses about how the decision is made. That means that 'when a decision is made by a complex expression that eventually evaluates to TRUE or FALSE, we want to make sure that each atomic condition is tested both ways, TRUE and FALSE.'. We therefore designed the condition coverage with a decision fork, and inside the decision, some atomic conditions, each to be tested. Something very important in this technique is to note that condition coverage does not automatically guarantee decision coverage.

[3] *Advanced Syllabus, ISTQB, Oct. 2007*

[4] *Advanced Software Testing Vol.1, Rex Black, 2009, Rocky Nook Inc.*

[5] *Advanced Software Testing Vol.3, Rex Black & Jamie L. Mitchell, 2011, Rocky Nook Inc.*

Thereafter comes a stronger coverage with decision/condition coverage, also called condition determination coverage. As it combines both previous coverages: each atomic condition and overall expression are to be tested both ways, we assembled both source ideograms in the resulting one, with a decision showed by the deep green box, including the atomic conditions represented by triangles.

After this, Modified Condition/Decision Coverage (MC/DC) is yet a stronger level of coverage. It adds one more factor: 'Each condition must affect the outcome decision independently while the other atomic conditions are held fixed'. As the number of tests to be designed for covering MC/DC is close to n+1 (with n atomic conditions), we separated the latter by a red '+'.

The last of all of the control-flow coverage schemes, which provides the highest coverage, is multiple condition coverage, where we test every possible combination of atomic conditions in a decision. As the number of tests is maximum 2n, before eventual short-circuiting, the chosen graphical relation between atomic conditions is a dark red 'X', for multiplying combinations.

Now the two last structure-based techniques are not based on control-flow coverage, but are related to path testing.

As explained by the ISTQB Glossary, the first path testing technique, the LCSAJ (Linear Code Sequence And Jump), consists of the following three items (conventionally identified by line numbers in the source code): the Start of the linear sequence of executable statements, the end (or Finish) of the linear sequence, and the target line (or Jump to) to which control flow is transferred at the end of the linear sequence. The ideogram for LCSAJ shows then: line numbers, S for starts, F for finishing and J for jumps. Two LCSAJ tests are exemplified.

The cyclomatic complexity, developed in 1976 by Thomas McCabe, helps us evaluating code complexity and the amount of test required to cover it. Each test is a unique independent path through the module – with no iterations loops allowed.' Three different basis paths are possible in the ideogram. As the number of basis paths is equal to the cyclomatic complexity, we highlighted in the ideogram the calculated cyclomatic complexity (in this case Cc = 2 'enclosed Regions' + 1 = 3).

Conclusion

MDV offers a new approach for the demonstration of test techniques. Firstly, the MDV Model provides a useful graphic way to spatialize, understand and memorize each test technique thanks to the specific feature of mandalas. Secondly the MDV Project could provide a practical and all-encompassing demonstration of the test techniques used within a single project.

MDVs are still under development (i.e. for project samples or reviews) and we would be pleased to share our models with you and to receive any feedback. Designing meaningful ideograms is quite a challenge for some techniques, so any complementary suggestions on these would also be most welcome.

Note: Special thanks to Mme Géninet for inspiring in me the MDV concept with her mandalas and having kindly allowed to reference her work in this article. Thanks also to Thierry Charles for his help in the first draft paper drawing of the model I had conceived, and thanks to Eileen Basgallop for her very kind help in correcting the English in this article.

And of course, special thanks to Damien, Franck and Grégory for all their support and significant help in challenging, improving and developing for tomorrow both MDV concept and this article.

> biography

Philippe Roux-Salembien
He has been involved in the services industry as developer, test support, test lead and test manager for 12 years. He has been engaged mainly in IT services for clients in banking, telecom and other industries. His involvement has primarily been in functional testing, having gained experience in TMAP NEXT®, also in training and support for some commercial tools. He has a French equivalent to a MSc. in computer science, and is an engineer in biotechnology in France, Switzerland and Germany. He is an ISTQB Certified Professional, and is interested in returning to an international role in the near future.

Damien Mathieu, Franck Launay and Grégory Heitz
The co-founders of FACTory Consulting are ISTQB Certified professionals. Damien and Franck are former network and system technicians. They've later come to software testing and now mainly concentrate on test analysts' tasks, training and testing tools usage and best practices. Gregory has worked for about 10 years as test manager and trainer in various companies (electronic banking, pharmaceutical and multimedia).