Automatic test data generation for complex forms in enterprise applications

The labyrinth and the dragon of contradictions

The more extensive and complex forms become in an application, the more hopeless is the manual provision of test data. mgm has therefore developed a procedure that generates test data for quality assurance during development in a fully automated way. A promising side-effect is that the procedure can also be used to detect contradictions in domain-oriented models.

This field may only be specified if *"it is a controlled company and a corresponding takeover profit has been declared in the individual details and the registration was made before 11.11.2019"*. These are the conditions for a single field of the 2019 Corporate Income Tax Form, a 197 page form, mind you.

The example gives an idea of how extensive and complex online forms of business applications can become. There are conditions and dependencies between the fields. And there are technical rules and interrelationships that determine which entries are allowed and which are not.

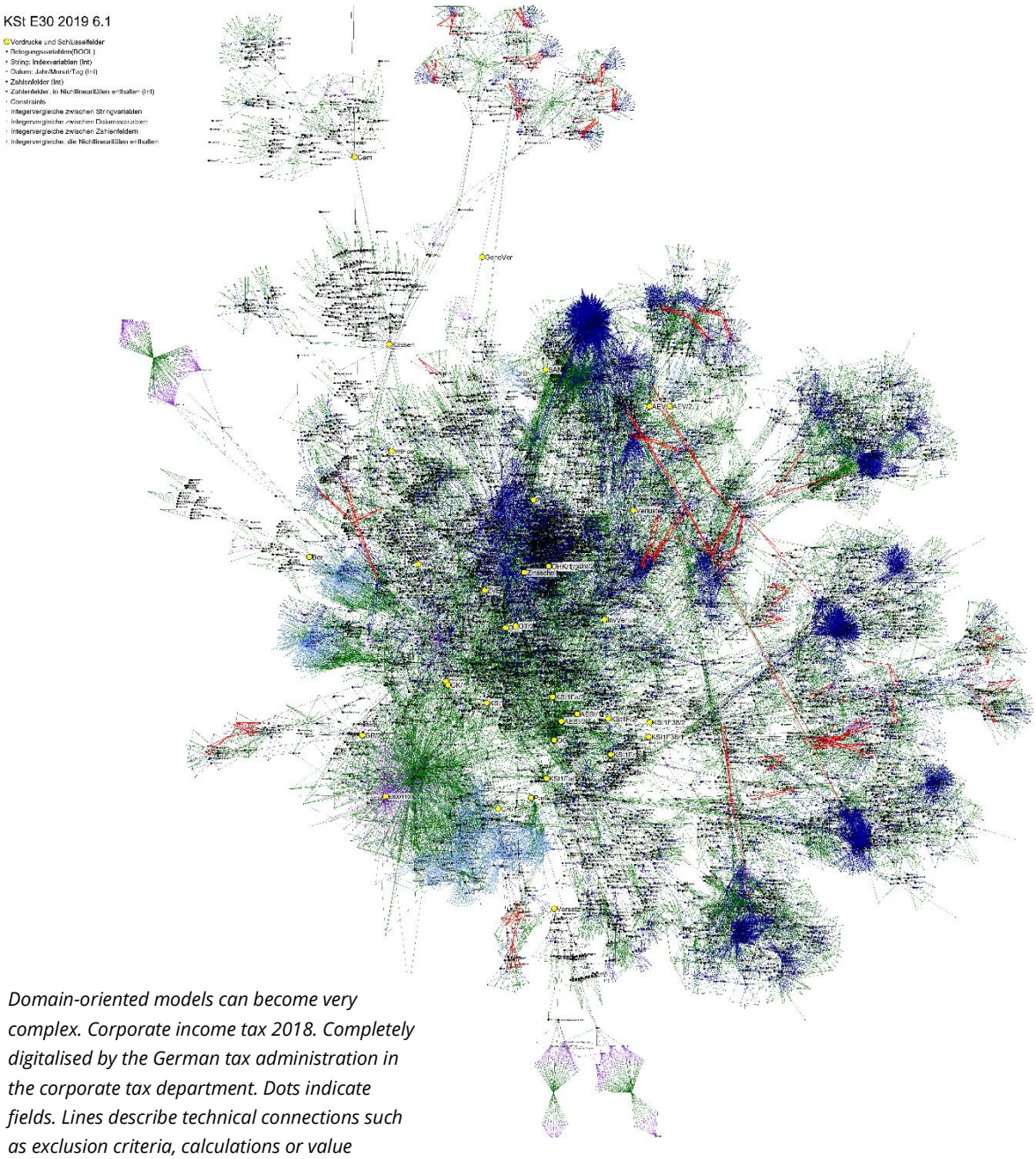
These are all challenges in the generation of test data - even if they do not have to be technically meaningful tax returns. Test data rather consists of random values. Or they sound out maximum values and check whether the input of all permitted characters works. And yet: how do you automatically find a valid data record for such a comprehensive form?

Thinking from the end

"We realised early on that we needed a generic solution," recalls Johannes Bauer. The graduate mathematician has headed the research project for test data generation at mgm for several years.

"Although we had been using some test data generation methods for years, we were not able to generate the test data. The methods did require specialist knowledge. This was always labour and time consuming."

Randomly generating test data was not an option. "For a very large form in a customer project, we calculated the probability of randomly finding a data set containing a valid combination of specified and unspecified fields. The result was 10^{-300} percent. The probability is astronomically low," says Bauer.



Domain-oriented models can become very complex. Corporate income tax 2018. Completely digitalised by the German tax administration in the corporate tax department. Dots indicate fields. Lines describe technical connections such as exclusion criteria, calculations or value comparisons.

Instead of approaching test data generation "from the front" - i.e. first generating data and then seeing if it fits - the approach starts "from the end". What must be fulfilled for a field to be allowed to be specified at all? The answer lies precisely in the domain-oriented A12 models of the model-driven approach. They represent the data specifications, validation rules and calculations behind the fields of a form. In other words: the domain-oriented application.

Decision problem: finding the right junctions

Even if the domain-oriented models are given, a number of challenges remain for the generation of valid test data sets. Many conditions defined for individual fields open up several possibilities for generating valid test data. One can imagine these possibilities as branches of a path. For each junction a decision has to be made which path to take. However, most combinations of branches create contradictions at some point, they lead to dead ends.

It is as if one of the few paths has to be found out of a gigantic labyrinth.

In one customer project, a very extensive technical model alone resulted in 212,000 possibilities. It is not nearly possible to try them all. We would walk through the labyrinth for half an eternity.

Ways out of the labyrinth: SMT-Solver

Fortunately, there are special algorithms that are specialized for exactly such problems.

They follow up the questions,

- about the order in which decisions should be taken,
- how to quickly find out which junction is a dead end
- and how to make use of previous knowledge in case of additional conditions and to which branches one has to go back.

These algorithms belong to the field of so-called "satisfiability modulo theories" (SMT) in computer science and mathematics. They deal with decision problems, where the solvability of logical formulas is at stake. There are special SMT solvers for practical use, which provide the algorithms in libraries. And there is a standardised language - SMT-LIB - for the form of the input of the models and the output of the data.

In order to use an SMT solver for the test data generation of applications developed with the A12 platform, a translation of the technical rules was first necessary. For this purpose, a bridge from the A12 rule language to SMT LIB was developed within the research project. The rules for a form are thus first translated to SMT LIB. The translation is chased through the SMT solver as input. The output of the SMT solver is then translated back into a data set in the format of the A12 rule language.

100% coverage of the fields

The test data generation is a very computationally intensive process. It resembles the search for a narrow paved path through a swamp area. The smallest misstep can result in infinite running time. Even though SMT solvers are constantly being improved, test data generation can take hours for very large models with complex rules. The optimal translation of A12 rules to SMT-LIB and the best setting and use of SMT solvers is the result of years of research and continuous improvement.

"We must invest regularly in improving performance. Business application models typically become successively more extensive and complex," says Bauer. "Today, we only need 20 per cent of the time to generate test data compared to previous solutions. And 100 per cent of the fields are covered. But the decisive advantage of the new process is that it is fully automated. No specialist knowledge of the contents of the forms is required for generation. We directly use the knowledge from the domain-oriented models".

A solution to the "multihopp" problem

"When developing the generator for test data generation, we realised that the process could solve another problem," says Bauer. "It detects inconsistencies in the rules which ensure that some fields cannot be specified."

A simple example: rule 1 states that fields A and B of a form must be indicated together. Rule 2 states that fields A and B must not be completed together. As a result, users cannot make entries for the fields concerned. They have been rendered unusable by the rule violation.

In practice, such simple contradictions stand out directly. However, there are clearly difficult cases which are very difficult to find. "We ourselves were amazed at how hidden some contradictions can be, which extend over several rules. In practice, for example, test data generation revealed a contradiction that extended over a connection of eight rules and eight fields in a tax form," says Bauer. "Internally, we refer to this phenomenon as a multi-hopping problem. A human being has hardly any chance of discovering such errors".

Identify contradictions directly during modelling

The developed method not only recognises that such contradictions exist. It also reveals which rules and fields are affected. "This information is invaluable for error-free modelling," says Bauer. "That is why we want to make it available to the business analysts and experts who create the rules as early as possible in the development process.

The procedure is currently being integrated into the modelling tools as part of the research and development of the Enterprise Low Code Platform A12. In the future, the tools will be able to point out such contradictions directly during domain-oriented modelling and ensure that the models are mathematically correct.

"It is a fine example of how the results of an internal research project can be put to secondary use," summarises Johannes Bauer. "In a way, we have not only found the ways out of the labyrinth. We have also tamed the dragons that live in the labyrinth.

mgm technology partners gmbh
Frankfurter Ring 105a
80807 Munich

Phone +49 89 / 358 680-0
info@mgm-tp.com
www.mgm-tp.com