

Domain-specific modelling for the Dutch Tax Agency: how MDE enabled civil servants to program tax law.

Tax law is complicated and requires complicated software for managing tax returns, collecting, and refunding tax as required at the scale of an entire country. As a result, changes to tax-management software are costly, take substantial amounts of time, and are potentially error prone. In a project co-led by Meinte Boersma from DSL Consultancy, a model-driven engineering approach to tax-management software helped to significantly reduce the cost of changes and increase the agility of the Dutch Tax Agency in responding consistently to changes in tax law. Where, in the traditional approach, tax laws are initially written by legal teams and require manual translation into software, the new system allows tax rule analysts to express tax rules directly in a language that is understandable to legal experts and can be automatically translated into an executable computer system. This enables an agile approach to tax law, which also supports the automated testing of tax-law rules for consistency and correctness.

What system was in place when you started the project?

The system was slow moving and ripe for an overhaul which would almost certainly lead to significant cost savings. The key contribution of MDE to the project is two-fold: the format that the domain (fiscal/legal) experts can understand effectively *is* the software's source code, and that source code is already testable (and tested) by the domain experts themselves, at that point. The source code is then only “transpiled” into Web services.

Tax law is typically refined into “tax regulation” which still takes the form of hand-written, natural language documents - typically as Word documents. These documents would then be given to the software developers who would translate them into source code. With Agile Law Execution Factory (ALEF) (the modelling tool), the original tax-regulation documents are turned directly into the specifications that implement the regulations. Before, these tax-regulation documents were first turned into the typical stack of functional and technical design and implementation documents. The latter doesn't happen at all anymore.

What was the objective of the project?

The objective was to create a software system that can be given to rule analysts (the people translating the tax law into rules). The system needed to be easy to use and provide confidence to the users that the output would be reliable and correct. The analysts then directly write down these rules into a very controlled form of Dutch. This would then be accurate enough and precise enough and complete enough to act as source code.

The team developed a domain-specific modelling language capturing this form of controlled Dutch for writing tax and taxation rules. Within the income tax department there are around 20,000 rules. Writing down the tax rules in this domain-specific modelling language allows the implementation of a web service to be automatically generated. This web service is the core of the Dutch Personal Income Tax management system.

What was it was about MDE that helped you make a difference in this case?

Rule analysts no longer need a software engineer on site. The use of a domain-specific modelling language enables rule analysts to formally capture their expertise in a computer-processable form, and the use of automated model transformations enables the automatic generation of executable software for tax management. Rule analysts are often people that have an accountancy background, or lawyers and are not computer developers. They do not have a background as programmers but the domain-specific modelling language enables them to enter, maintain, and manipulate the taxation rules and regenerate the final software system.

The domain-specific modelling language further supports the definition of test cases that can be automatically executed. This was not possible with the previous highly manual process. A rule is written down, along with example data and executing the tests compares the results produced by the taxation rules with the expected results encoded in the test data. The domain-specific modelling language supports continuous integration and automatically executes all tests as a regression test suite. This ensures that the rules written down work and work correctly.

What were the challenges along the way and how were they handled?

The biggest challenge was that an organisation like a tax agency is very risk averse and is used to systems and processes used over many years. Once the tool was created it was supplied to staff at the Tax Agency to try it out. The user could then try out the tool and see that the process worked much more efficiently than the current process in place. The end product would be something the users own themselves. Before ALEF was created, it was difficult to relate back to the rules what the code implements. Source code was in use within the department and at that time, physical rooms for up to eight developers were required in order to maintain and debug the work. Due to the introduction of the tool, the maintenance and debugging could be completed by the rule analysts. The introduction of ALEF empowers the agency staff and means developers are not relied upon in the same way in which they were before.

Using the tool is also incredibly cost-effective because it is so much quicker than the old system. Apart from creating the tool at the outset, the development involvement has been removed.

What are the next steps for the project?

The tool is currently used within the Income Tax department, which accounts for around eight million filings annually. There are also a further 83 other areas of tax law which have some kind of automatic processing already so there is real potential for further extension of the scope of the domain-specific modelling language.

Additional information:

- <https://resources.jetbrains.com/storage/products/mps/docs/MPS DTO Case Study.pdf>
- <https://staf2019.win.tue.nl/wp-content/uploads/2019/09/STAF-ID-Adema.pdf>
- <https://werken.belastingdienst.nl/nieuws-en-artikelen/software-maken-met-een-fabriek-115>
- <https://www.werkenvoornederland.nl/organisaties/belastingdienst/nieuwe-geautomatiseerde-software-bouwen>

- <https://gitlab.com/commonground/virtueel-inkomstenloket/regelbeheer>

Meinte Boersma is the independent consultant at DSL Consultancy visit: [dslconsultancy.com](https://www.dslconsultancy.com)

Meinte was interviewed by Nuno Ferreira da Silva, Network Coordinator of MDENet. For more information about the network visit: [mde-network.com](https://www.mde-network.com)