August 2022 | #3 edition

MDENET

The home of model-driven engineering

MDENet | The home of model-driven engineering

NEWSLETTER

#3 August Edition 2022

On this edition:

MDE Newsletter - Industry Stories

Newsletter by Joost Noppen

Model-driven engineering and Model-based engineering has long held great promise for the development of software systems. But due the initial learning curve and the myriad of approaches and techniques it can be hard to envision how best to apply the principles and building a product and company around it. To provide some inspiration and ideas, in this newsletter we talk with two MDE practitioners who have made the principles and tools an integral part of their work. First we talk to Pedro J. Molina, founder of MetaDev and a daily practitioner of Model-Based Development and Model-Driven Engineering. Second we talk to Machiel van der Bijl who is CEO-founder of Axini and we talk with him about Model Based Testing and the Axini platform.

In addition we cover the highlights of the MDENet Thematic Workshop on Systems Engineering which took place on the 11th of May of this year.

METADEV

PEDRO J. MOLINA

Our first practitioner we interviewed is Pedro J. Molina who is founder of Metadev and member of the OpenAPI Initiative.

Pedro, can you tell us a little bit about what kind of problems you work on?

We help create software for our customers, specifically focussing on cloud strategy and architecture definition. Personally I am particularly interested in the practical application of model-driven engineering and domain specific languages to reduce repetition in source code and to ensure all the business logic is put in one place.

What problems have you encountered where model-based development and model driven engineering could be helpful?

A very typical problem we face is helping customers create a family of products in a specific domain, with common elements and variability. In addition to realising and supporting reusable common elements, we apply model-driven engineering in the variability part of this kind of challenge. We typically create a domain-specific language to describe the variability aspects of the software system, which in turn allows us to code-generate the areas of the product family where this variability is needed. This kind of approach in its concept is similar to what Software Product Lines are trying to achieve, however the use of DSLs and MDE allows for more range and flexibility which is beneficial for our customers.

"I AM PARTICULARLY INTERESTED IN THE PRACTICAL APPLICATION OF MODEL-DRIVEN ENGINEERING AND DOMAIN SPECIFIC LANGUAGES TO REDUCE REPETITION IN SOURCE CODE"

METADEV

PEDRO J. MOLINA

And how has model-based development and MDE been helpful?

The main benefit of using model-driven engineering and model-based development is that you have a single source of truth. The model is the place for your updates and maintenance, everything else follows from that. And in our case that is true across a family of products, which leads to substantial benefits. For example, it is possible to enforce policies of auditing and compliance in the code generator and then use the model to define the specifics of compliance in a domain. Once you certify the code generator this way, it is no longer necessary to certify individual products within the family, which drastically reduces lead time and time to market.

And are there any downsides?

Everything has its downside, MDE is no different. Fundamentally it is another level of indirection which means you cannot just fix code. Rather, you have to fix the model and regenerate the code and make the translations from problem to solution. Testing itself is not as big a problem as you might think. When using a code generator, you can also generate tests for the generated code. The problem is when you also write manual code, in which case you will need to have full coverage unit tests and more importantly a lot of integration testing.

What would be your advice for a company who is facing similar problems and is about to embark on their first steps with MBD and MDE?

I would advise to first read about the state of the art, and once you have taken in the basics try to define the language you want to go with at the right level of abstraction. And start building as soon as possible and iterate. As with software technology it is about getting hands-on to find out what works.

"THE MAIN BENEFIT OF USING MODEL-DRIVEN ENGINEERING AND MODEL-BASED DEVELOPMENT IS THAT YOU HAVE A SINGLE SOURCE OF TRUTH."

AXINI

MACHIEL VAN DER BIJL

Second we spoke to Machiel van der Bijl, founder of Axini, a company that specialises in test automation through the application of model-based testing.

Machiel, can you tell us a little bit about what kind of problems you work on?

Within Axini we work on test optimisation and test automation. Specifically we build on principles of model-based testing, which allows you to specify the requirements of a software system which our platform can verify without having to write tests. This means our customers can comprehensively test and verify their software in less time than traditional test approaches. Our focus is on the realisation and expansion of our platform and modeling environment that allows customers to specify and test their requirements against their implementation.

What problems have you encountered where model-based development and model driven engineering could be helpful?

Our core application area for model-driven engineering and model-based testing lies in the code generation for tests based on the requirements specification provided by the customer. The requirements specification is done using a kind of domain-specific language that in turn can be interpreted and used for generating many different test-cases. An intuitive example is a requirement specifies a division of numeric values that need to be performed. Our platform understands this kind of operation which means the code generator can create tests based on division-specific edge cases, like division by 0.

The requirements model we provide can be used incrementally, so it is not necessary to provide a full specification up-front. However it is possible to reason across the entire specification that is available to for example determine consistency and completeness, which is a benefit you would not get from unit testing.

"WE BUILD ON PRINCIPLES OF MODEL-BASED TESTING, WHICH ALLOWS YOU TO SPECIFY THE REQUIREMENTS OF A SOFTWARE SYSTEM WHICH OUR PLATFORM CAN VERIFY WITHOUT HAVING TO WRITE TESTS. "

AXINI

MACHIEL VAN DER BIJL

And how has model-based development and MDE been helpful?

The clearest benefit is the ability to generate many test cases for software systems that would be impossible to write manually. However the creation of such requirements specification also opens the door for additional benefits for our customers. The underlying formal principles of our modeling platform and code generation have significantly streamlined software audits for our customers. As there now is a single source of truth it also becomes easier to keep on top of the system itself and integrate with other systems.

Are there any downsides?

This kind of approach is less suitable for GUIs as they tend to be too slow for the numbers of tests we can generate. And this is something worth considering in a wider sense when working with MDE. There are areas where the high levels of automation that can be achieved do not necessarily work. As with all software development there are trade-offs and suitability concerns. A second consideration is that the application of our approach and MDE in general will require a different way of working as your development process becomes more front-loaded while banking on a shorter tail-end of the process. Your model becomes your main deliverable, which means your process and organisation needs to centre around it.

What would be your advice for a company who is facing similar problems and is about to embark on their first steps with MBD and MDE?

For customers to get involved in our way of working we start with understanding what the problem is they are trying to solve, followed by a short demo to help understand what is possible and what the benefits are. This principle really applies to using model-driven engineering and model-based development in general.

This is also reflected in our own process where we try to get hands-on as soon as possible. We typically have a 5 day quick-scan followed by a 4 week proof-of-concept to get to a small implementation that clients can build on. Most clients want to see our platform and approach work in their environment so it is important we get there as soon as possible.

"THE CLEAREST BENEFIT IS THE ABILITY TO GENERATE MANY TEST CASES FOR SOFTWARE SYSTEMS THAT WOULD BE IMPOSSIBLE TO WRITE MANUALLY."

Thematic Workshop on Systems Engineering

On May 11, 2022, MDENet held its second thematic workshop, focusing on systems engineering. MDENet aims to bring together researchers and practitioners with an interest in MDE, either as experts in the technology or as stakeholders interested in the use of MDE. A key aim of MDENet is to support the creation of new collaboration between MDE experts and experts in other areas by identifying opportunities for cross-pollination. One mechanism towards this are MDENet's thematic workshops

The workshop on Systems Engineering started with a keynote and brief introductions of participants' backgrounds, but the majority of the time was focused on discussions aiming to elicit emerging themes for potential new collaborations.

The workshop was well attended by researchers and practitioners from academia and industry. Discussions were very lively, and if anything there was not enough time to complete all the discussions of interest to participants. You can find a full report of the workshop and the intended next steps here: https://mde-network.com/wp-content/uploads/2022/07/Report-from-the-MDENet-thematic-workshop-on-Systems-Engineering.pdf

"DISCUSSIONS WERE VERY LIVELY, AND IF ANYTHING THERE WAS NOT ENOUGH TIME TO COMPLETE ALL THE DISCUSSIONS OF INTEREST TO PARTICIPANTS."

Coming up this year

The most notable upcoming event is our Annual Symposium, on December 1st and 2nd 2022. The programme is still being finalised as I write, but there will be something for everyone: there will be a range of keynote talks, industry sessions, seedcorn project talks, as well as hands-on tutorials and challenge talks. Keep an eye for more details in the coming months.

There will also, of course, be new research demonstrations, training events, thematic workshops and more. Something you'd especially like to see, or to offer? Come and post on the platform about it!