JetBrains MPS Community Meetup 2019: Videos and Slides

MPS Community Meetup 2018: Videos and Slides

Tackling Security and Safety with MPS by Klaus Birken from itemis (SLIDES and VIDEO)

As the complexity of technical systems is rising, security and functional safety are two cross-topics which are crucial for the success of a product. Appropriate tools are needed to cope with the challenges resulting from those topics, applying mechanisms like abstraction and automation. MPS is a proper platform for building such tools. In the talk, we will demonstrate two solutions itemis has built in order to help customers tackling security and functional safety problems, and how the capabilities and flexibility of MPS support this. These two solutions cover the range from modeling and analyzing distributed systems to weaving patterns into C-code, from end-user RCP product to research prototype, from graphical editors to shadow model transformations.

Simplified Packaging And Publishing of MPS Plugins by Sergej Koščejev (MPS Consultant) (SLIDES and VIDEO)

Compared to platforms like Java, JavaScript, or Ruby, packaging, publishing, and using MPS plugins involves more effort, or "friction." To save on the effort to build the necessary build scripts, plugins are merged into large platforms (MPS extensions, mbeddr.platform) and the downstream projects have no choice but to depend on coarse-grained dependencies. I am convinced that this hurts the MPS plugins ecosystem and I've started working on reducing this friction. In this talk, I want to demonstrate my progress so far.

DSLs with MPS: Big design upfront? by Kolja Dummann from itemis (VIDEO)

As the whole software industry has moved to agile processes we also develop languages in such environments. How can we facilitate agile methods and develop better languages and tools around them? Based on severals past and present projects DSLs and Tools based on MPS in an agile way. What are the strong points of MPS in such a context and challenges did we encounter on the way? In addition, we want to see how iterative language development affects language.

Embedded software modeling using Capital Software Designer by Dinesh Kumar Rajamani and Jan Richter from Siemens (SLIDES and VIDEO)

Capital Software Designer (CSD) is a solution from Siemens for onboarding embedded software, which drives integration, verification, and validation activities. It is part of the capital enterprise solution for defining, designing, producing and maintaining embedded systems across electrical, electronic, and software aspects. Capital Software Designer is built on the JetBrains MPS language workbench. In this presentation, we will use CSD Designer's positioning and capabilities, and highlight the following. We leverage the projectional editing capabilities of MPS to allow the users to model in textual or graphical notation. The real-time architecture import model comparison helps architects to merge the architecture created from different standards such as SysML, AADL, and Autosar. The APIs exposed for CSD help in accessing and creating the MPS model from any outside environment, thus making MPS a service.

Normalized Natural Language: The Case of Specification and Verification of Product Behavior by Jennek Geels (SLIDES and VIDEO)

KISS is a method to model system behavior using concepts from natural language. These results in models are both human-readable and meaningful. The intention of the KISS method is to make all natural language executable, but to define a small executable subset using normalization rules. This method was created in the Netherlands in the 1990s and it has been used intermittently in our R & D. This year it is applied to Specification and Verification of Product Behavior. We have JetBrains MPS to build a model editor with generators to .docx for specification agreement and generators to SpecFlow / Karate / Robot for automated verification. In this talk I want to do the following:

- Give a brief introduction to Normalized Natural Language and how to define it.
- Show its use in Specification and Verification of Product Behavior
- Demonstrate the status and roadmap of the MPS implementation

New Typesystem Aspect: Path to Expressive and Natural Typesystems by Grigorii Kirgizov from the MPS team (SLIDES and VIDEO)

One of the biggest upcoming features of MPS is the typesystem language - a powerful new tool for language designers. With its help, advanced typesystems with features such as type inference or generic types can be easily and simply described.

In the presentation, we will touch upon the strong logical foundation of the new language and tell you about the motivation behind the current typesystem aspect. We want to talk about our plan to smoothly transition from the legacy aspect to the new one, highlighting the most interesting features of the language, and, of course, show some examples.

MPS Keynote (SLIDES: MKT Insights, What's new in MPS and Web MPS, VIDEO)

There are a lot of things happening around MPS that we are eager to share with the community.

Since the last community meetup, we've been looking at the major issues and we want to tell you all about the recently added features. Our community has grown a lot too, and there are insights we want to share about that. So we want to talk about the services we offer to help and support our customers in their MPS projects.

It's not a secret we've been working on Web MPS, and now we're ready to demo this technology and explore its future with you. Finally, we want to outline the roadmap for MPS going forward.

Finally, a declarative base language for MPS! by Wim Bast from Modeling Value Group (SLIDES and VIDEO)

We want to give a demo of DclareForMPS. DclareForMPS is open source plugin for MPS. DclareForMPS adds a language aspect named 'rules' to MPS. The rules are defined in the regular MPS base-language. Dclare has reactive and incremental semantics. You can build transformations that synchronize your models in real-time. You can therefore use DclareForMPS to derive only a portion of an existing model within itself. We want to demo a transformation between two models of two different languages for a sudoku solver in MPS. The demo wants to be given by Wim Bast of the Modeling Value Group - an early advocate of model-driven development and an experienced MPS user.

Taming the Computed Tomography Scanners Complexity with Domain-Specific Languages by Holger Nehls from Siemens Healthineers (SLIDES and VIDEO)

Using the potential of projectional editors to switch from generic spreadsheets to a workflow-driven application

Four years ago, Siemens Healthineers started a project on the complexity of computed tomography (CT) scanners. The result is a domain-specific modeling tool, build with MPS, that replaces the traditional documents based approach for scanners specification.

During the productive use of the first version of our application, we learned that this is a very good idea. This leads to the completion of a workflow-driven application that benefits from the unique capabilities of MPS.

In the talk, I want to present the result by a demo and summarize the challenges we faced.

The second part is about our Continuous Integration (CI) chain and the (ongoing) implementation of DevOps principles for the domain-specific tool.

Are Tax Specialists the New Programmers? by Diederik Dulfer from Dutch Tax Office (SLIDES and VIDEO)

At the Dutch tax administration, we have used MPS to create a tool that supports our DSL RuleSpraak. This language is a controlled natural language to specify the rules for tax calculations. The Dutch Tax Administration uses this language to implement the Anti-Tax Avoidance Directive (ATAD), which contains five legally-binding anti-abuse measures. In the process of specifying rules for the anti-tax avoidance directive, we work closely with tax specialists to specify the rules and the test cases. IT is no longer a black box for them. With the controlled natural language, they are themselves in the driving seat of IT development. And, of course, they know better than we do. In this session, we will talk about this project and other projects where we use MPS to support our development.

Workday's transformation to a new language platform: Developing a full-fledged programming language with MPS by Bircan Copur, Antonio Gliubich and Alexey Yashin from Workday (SLIDES and VIDEO)

Some time ago at Workday, we set out to supplement and eventually replace our in-house, web-based metadata programming language XpressO with a more modern sibling, YP. We set out to create a DSL, but soon found ourselves on the road to developing a general purpose programming language using MPS. In this talk we will discuss how we tackle interoperability with

10+ years of legacy XO code and dive into some specifics such as how we try to solve different problems: from updating instances in an immutable programming language, to translating all kinds of text strings into multiple natural languages, as well as implementing generic types with the help of the MPS typesystem.

Dependency Modules by Kemal Soysal from LS IT-Solutions GmbH (VIDEO)

In MPS, as dependencies, we use languages, modules, models, and java stubs from .jar files, among other things in a model. Although we can write our own plugins, we may sometimes need to use plugins from other vendors. We need to be familiar with our MPS or User Config Folder to open a project. By modeling each dependency in the project itself, we can:
* Adjust the version, type, and location information.
* Integrate into the logical view.
* Load / unload the dependency.
* Introspect its dependencies.
* Navigate to the source or other artifacts.
* Support automatic building.
The core of dependency modules is extensible, allowing users to extend it to their needs.

GDF: a Gamification Design Framework powered by Model-Driven Engineering by Antonio Bucchiarone from Fondazione Bruno Kessler FBK (SLIDES and VIDEO)

Gamification refers to the exploitation of gaming mechanisms for serious purposes, like promoting behavioral changes, soliciting participation and engagement in activities, and so on. In this talk, we present the Gamification Design Framework (GDF), a tool for designing through model-driven engineering mechanisms. In particular, the framework is based on a set of well-defined modeling layers that start from the definition of the main gamification elements, followed by the specification on how these elements are composed to design games, and then progressively to reach concrete game implementation and execution. The layers are interconnected through specialization / generalization relationships. The approach has been validated by means of JetBrains MPS and has been validated through a gameful system in the education domain. The opportunity to reduce the complexity of the game by the other means the opportunity of reusing portions of a game in other scenarios. Moreover, it enables the evolution of the concept and the evolution of the situation.