

## Project Note

# Highest Flexibility without Compromise

## Model Driven Development with IBM® Rational® Rhapsody® – Example of Machine Manufacturing

High quality, flexibility and user friendliness distinguish the wire processing machines of the Schleuniger Group. To cope with these challenges, the machine manufacturer from Thun is focusing on a platform strategy and Model Driven Development (MDD) with Rational Rhapsody. A discussion with Thomas Nyffenegger, Markus Walser and Sebastian Becker, Software Engineering, Schleuniger AG.



[www.schleuniger.com](http://www.schleuniger.com)



### PowerStrip 9550

The powerful cut & strip machine distinguishes itself by highest flexibility.

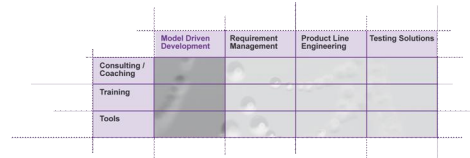
The Schleuniger Group, a business division of the Metall Zug Group, is a worldwide leading supplier of innovative wire processing machines. At its headquarters in Thun, Switzerland, electronic, mechanic and software components are developed for highly competitive cut & strip machines. Driven by the rising demand for quality and flexibility of the fully automatic machines, Schleuniger decided in 2004 to introduce Rational Rhapsody Developer in the software development.

### Modularity – Critical for Success

Schleuniger's customers include companies from the areas of telecommunications, machine and system engineering, electronics, MedTech, military and aviation. "However, a large part of our machines is used by suppliers of the automotive industry", Thomas Nyffenegger explains. "For the wire harness manufacturers user friendliness and flexibility are most important, beside a high quality of the processed wires of course. It must be possible to retrofit the machines according to demand – and the changeover must be fast and simple." To accomplish the development of the new machine generation Schleuniger focused on modular construction, intuitive operation and reusable platforms.

### Reusable Platforms

In autumn 2010 the PowerStrip 9550 was launched in the market – the second cut & strip machine which is based on Rational Rhapsody developed platforms. "It's state and sequence diagrams were decisive and although even today many algorithms and a very dynamic behaviour are behind the diagrams, Rational Rhapsody simplified development massively for us", explains Markus Walser.



**Thomas Nyffenegger,  
Schleuniger AG**

„Thanks to the fully automatic code generation with Rhapsody® we could accelerate the software development.“

**Advantages are Convincing**

The introduction of Rational Rhapsody and the development of the first platforms lasted about two years. “We had to master not only a new tool and a new programming language, but also integrate existing processes“, Thomas Nyffenegger comments on the lead time. “By doing so, UML and the visual presentation of dependencies were very useful for us.“ Other advantages Schleuniger lists are:

- coping with the increasing complexity is easier
- cost savings thanks to reusable platforms
- shorter Time-to-Market
- easier integration of new functionalities.

**Lessons Learned**

“I would recommend Rational Rhapsody any time for development tasks in the area of embedded systems“, Sebastian Becker reinforces. “The communication is easier, the documentation is always up-to-date and the class and sequence diagrams are very convenient“. However, target oriented trainings during the introduction and on-the-job as well as demand-oriented coaching are important, complements Thomas Nyffenegger. “One must know the tool, the philosophy behind it should be understood and one should think about the set up of the design right from the start.“

**Potential for Improvement**

Thomas Nyffenegger could imagine very well developing even more components with Rational Rhapsody. “A uniform basis and the graphic illustration are making discussions substantially easier“. Sebastian Becker sees potential for improvement above all in the user friendliness and in the handling of properties. “The binding of the version management and the administration of several development branches are not at optimum yet“, adds Markus Walser.

**Used Technologies**

Development Environments	Before Rational Rhapsody	With Rational Rhapsody
UML	Rational Rose	Rhapsody® Developer
IDE	DAC	Visual Studio
RTOS	None	Windows CE
Zielplattform	Motorola 68332	XScale
Programmiersprache	C	C++
Debugger	Hi-Wave	Visual Studio