IAC-04-U.3.b.03

THE SUCCESSFUL DEVELOPMENT PROCESS WITH MATLAB SIMULINK IN THE FRAMEWORK OF ESA'S ATV PROJECT

Christian D. Bodemann

ATV Project Manager
VEGA IT GmbH, Darmstadt, Germany
christian.bodemann@vega.de

Filippo De Rose

ATV Simulations Model Developer
VEGA IT GmbH, Darmstadt, Germany
filippo.derose@vega.de

In the development of spacecraft simulators in the past and in the present numerous different approaches, technologies and tools have been used. However a good tool or technology alone does not make a project successful. The success of a project depends on the harmony of management, the team and the tool. The aim of this paper is to present the successful usage of the Matlab / SIMULINK* in the simulators of the ATV (Automated Transfer Vehicle) project. The paper also aims to describe how SIMULINK supported the teamwork and made the development process more transparent. Form the experience of the ATV project it can be stated that SIMULINK has proved to be a very powerful development tool for real-time simulators. With the options it offers it can support the team and the management in making a software development successful. If SIMULINK is used efficiently it can make the difference, but it is like any other powerful tool or technique, only if used with care and discipline it will bring success. In the ATV project this was the case.

INTRODUCTION

The aim of this paper is to present the successful usage of the Matlab / SIMULINK in the simulators of the ATV (Automated Transfer Vehicle) project. But a good tool alone does not make a project successful. The paper also aims to describe how SIMULINK supported the teamwork and made the development process more transparent.

The Automated Transport Vehicle ATV is one of the contributions of Europe for the International Space Station. Throughout the development of ATV a number of non real time and real time simulators are used to validate the design, the functional validation of the ATV flight software and the functional qualification of the ATV vehicle. This qualification consists in verifying the on board software integrated in a complete system environment functionality. (Hardware in the loop) For the real Time Simulators

* Matlab, Simulink, Real-Time Workshop are trademarks of the Math Works, Inc. Eurosim is a trademark of Dutch Space. Mosaic is a trademark of NLR

Matlab / Simulink had been chosen as development tool to generate so called numerical models. The development of the ATV numerical models is based on the classical V cycle. For the development SIMULINK has been chosen and EUROSIM as the Simulator Platform. The SIMULINK and EUROSIM tools cover all the cycle activities:

- SIMULINK was used during the specification and design phases of the numerical models, the coding phase and during the models validation phases on host machine.
- Real Time Workshop was used for generating ANSI C code and Mosaic was used during the integration of the models into EUROSIM.
- EUROSIM was used for the design of the simulator, for the real time integration of the numerical models and finally for the validation of the overall simulation software.

During the 48-month of development the models had to be many times adapted to the needs of the different kinds of simulators or changes of the design of the item to model. Due to the chosen approach this could be done rapidly without severe problems. SIMULINK was never before used for a Simulator of such size and complexity, so all techniques of designing and programming had to be developed. Although this was not a classical object oriented design approach and it is auto generated C code, all models developed run within or faster than their hard real time performance limits. As SIMULINK was used in all phases of the project, the amount of different programming languages and techniques could be limited to a minimum. This made the integration of new team members faster and less expensive. The library-based design of the SIMULINK models supported the reuse of code within the current and future projects. As all models could be delivered in time and the amount of software problem reports is minimal, the project can be considered as successful. If a project was successful or not, in any case the tool used to do the development has a major influence on the project success.

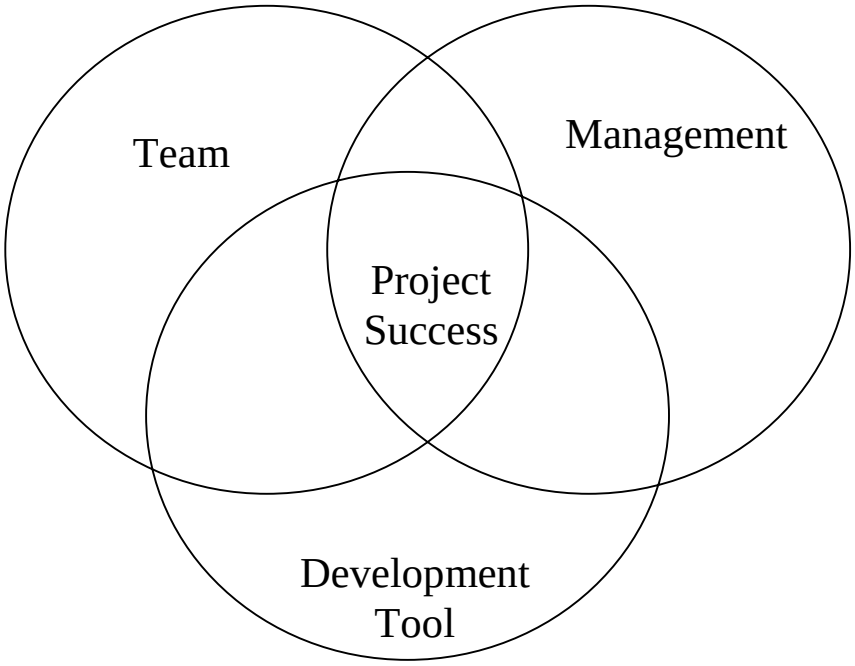

Fig. 1 Software project success dependencies

SOFTWARE DEVELOPMENT DEPENDENCIES

Every Software project is dependent on three major aspects, the tool used to do the development, the teamwork and the management approach.

If a project shall be successful, all the three aspects must harmonise. A team can not be successful if the tools used are not usable to do the work. In many projects tools were used, which promised to make the work easier and faster to do, but during the project it turned out that they were not maiden enough or too complex. On the other hand it is nearly impossible to manage a project if the tool does not allow teamwork. A tool, which does not allow or support teamwork, might work quite well at the beginning, but when it comes to the integration of the different software parts many projects faced unwanted surprises. So in order to ensure a successful software development a tool should support the following:

- the Software development
- the Teamwork
- the Management

As the ATV model development can be considered as very successful in the following it will be shown why and how SIMULINK fulfilled the, above described, expectations.

How did SIMULINK support the software development?

Normally the design of the models would be developed using a CASE (Computer Aided Software Engineering) tool supporting an object-oriented methodology such as the Unified Modelling Language (UML), an object-based methodology such as HOOD, or a structured design methodology such as Yourdon. These tools generate the model stubs and the models would then typically be implemented by hand in C++, ADA, or C.

For ATV the MATLAB SIMULINK tool has been selected for the model specification, design and development[t]. Although it does not follow a particular methodology, it offers most of the benefits of a CASE tool and also supports the execution of the models within it. More over Simulink has to offer a lot more:

- The schematics produced for the technical specification, describing the model, are already the skeleton of the code.
- The schematics have been reused in the architectural design document. The ADD focused in the case of ATV on the description of the interfaces to external models and internal interfaces. As the SIMULINK schematic was the baseline for coding all interfaces could be already identified and defined.
- As the SIMULINK is based on these schematics, the model was easy to maintain.
- Visualisation of results was easy and very helpful for debugging.
- Interpreter helped to avoid bugs.
- Good documentation with report generator.

The above mentioned tools and techniques can not or just partially offer this. Most of these are only used during the software design phase and then not anymore during the lifecycle of the software.

How did SIMULINK support the teamwork?

As good teamwork can be considered if every member of the team works on a different part of the software, but they exchange experience, knowledge and code. It can be also considered that, if a team works well, it is relatively easy to maintain coding standards and achieve a high level of quality

[t] MATLAB SIMULINK was chosen mainly because of a tool called Real-Time Workshop. With this tool ANSI C code can be auto-generated. This code can then be integrated into any real-time simulation environment.

without duplication of work and effort. With SIMULINK it was possible to come very close to this as:

- for all model testing more less the same test approach for every model could be used due to nature of the design approach,
- it was possible to reuse of many functional oriented designed models in different subsystems,
- It was possible to reuse the test harness. As the test approach was identical a common scheduler for testing the models could be used. So it was easy to reuse parts of tests from other models.
- The ADD was defining the interfaces, fewer problems occurred during the integration of models, written by different developers.

How did SIMULINK support the management?

For the management the tool used to develop the software should help to keep control of the development process and development time. Every tool, which also shortens development effort, ensures high quality and does have a high training need, can be considered as helpful. In the case of SIMULINK many expectations could be fulfilled as:

- The schematics of the technical specification and ADD are already code. This helped to keep an overview of the development process.
- ADD defined clearly the interfaces. This helped to avoid integration problems.
- The Report generator of MATLAB /SIMULINK shortened the time to generate documentation of test results.
- a generic test approach could be used, which lowered the effort of testing,
- The Visualisation of test results which helped to ensure the high quality.
- A short time needed to train new team members. New team members could be trained within two weeks. After this they could be integrated in the development process.

Reading all the above it could be assumed SIMULINK is simply perfect; in fact it isn't, it is just a very good tool.

Where did SIMULINK not support the project?

Assuming the ideal tool described above, there are some areas were SIMULINK does not support a software project.

1. MATLAB / SIMULINK is quite expensive. For a big project like ATV the licence costs have been considered as affordable, but for a small project an analysis of costs should be done.

2. As any other tool on the market there are update versions of SIMULINK offered on a regular base. Although there should be no version conflicts, during the ATV project some occurred.

3. SIMULINK can not use STRING variables. Therefore logging is more complex than needed.

4. The Memory concept of MATLAB / SIMULINK is not supporting the needs of a real software project. The concept used is good for scientific numerical or mathematical analysis simulations but not really for a real time simulator model. Therefor an initialisation routine had to be developed.

5. MATLAB / SIMULINK does not follow a particular methodology. Therefore a design approach and coding standards had to be developed.

6. Sometimes Developers coming from a text based development environment have some difficulties to get used to work with a graphical development environment.

CONCLUSION

For the ATV test facilities simulators a number of numerical models were needed. MATLAB / SIMULINK had been chosen as the tool for this task. It offered the unique possibility to be used during nearly all phases of the development. In the case of the numerical models for the ATV test facilities it is used for specification, design and coding. The code generated with Real-Time Workshop had been successfully integrated in EuroSim. During the first integration a number of technical problems were found, but all were solved.

Although MATLAB SIMULINK is a CASE tool, which is not object-oriented, a function-oriented design could be developed, which offers nearly all the possibilities of a classical object-oriented approach.

For the purpose of usage in hard real-time simulators MATLAB SIMULINK demonstrated many capabilities. Although it produces auto-generated C code, all models developed so far have run within or faster than their hard real-time performance limits. The transfer between different platforms like SUN / Solaris, PC / Windows and SGI / Unix never caused additional problems other than the length of names.

SIMULINK has proved to be a very powerful development tool for real-time simulators. With the options it offers it can support the team and the management in making a software development successful. If SIMULINK is used efficiently it can make the difference, but it is like any other powerful tool or techniques, only if used with care and discipline it will bring the success. A good tool can not replace good management or efficient teamwork but it can make it easier to manage and possible to establish real teamwork. On the other hand, without a good tool, like SIMULINK, it is far more difficult to have success.

The numerical models for the ATV test facilities simulators are developed by an EADS Launch Vehicles, DATASPAZIO and VEGA integrated team at EADS ST premises in Les Mureaux / France.

LITERATURE

[1] Bodemann, C. D., De Luca, A., Mauro, M., Bianco, F. : ATV Test Facilities Numerical Models. DASIA Conference, Nice, June 2001

[2] Mauro, M., Bianco, F., Bodemann, C. D., De Luca, A., Guillard, V.: SIMULINK as a development tool for Real-time simulators: the ATV Test Facilities experience, DASIA Conference, Dublin, May 2002

[3] De Luca, A., Bodemann, C. D., Grasselli, U., Schirone, L.: Power System Simulators for Design and Operations: Two different approaches?, ESPC Conference, Porto, May 2002

[4] Herold, H., Klar, M., Klar, S.: Objektorientierung, Angewandte Objektorientierung mit C++ und UML. Addison-Wesley Verlag, 2001

[5] Bodemann, C. D., De Luca, A., Mauro, M., Bianco, F. : Function Oriented Model Development for Real Time Simulators with MATLAB / SIMULINK. AIAA Conference, Monterey, August 2002