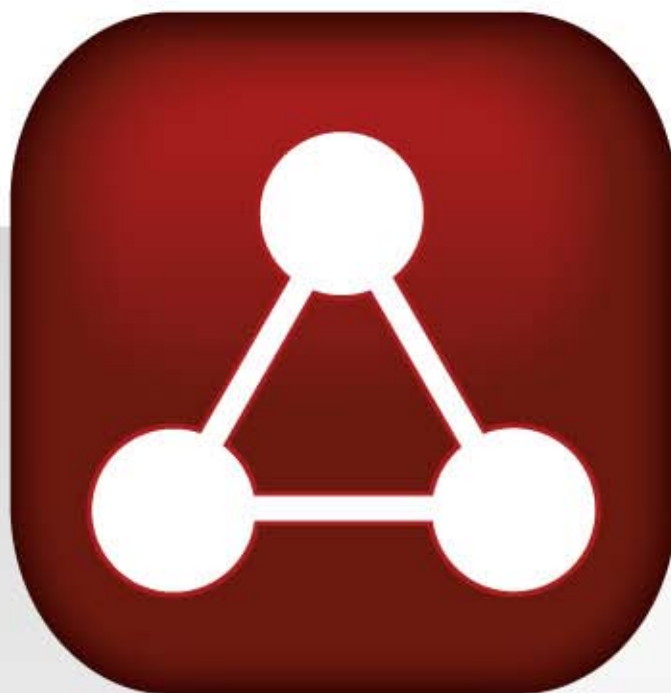


Improving productivity and quality using the ASD:Suite

Sioux evaluates Verum's ASD for embedded software

Authors: Leon Bouwmeester and Arjen Klomp

Version: 1.0 21 September 2009



The simple way to build
complex software
systems

Executive summary

For this Analytical Software Design (ASD)¹ project, Sioux's Embedded Systems group set out to assess the improvement in software development time, cost, and quality that could be achieved when using Verum's ASD:Suite. Previously Sioux had developed a software controller for a hot food vending machine. The purpose of this project was to use ASD:Suite to model the behaviour of the software controller and generate C# code, then compare development performance with existing metrics.

Sioux, based in The Netherlands, Belgium and Russia, supplies software development services for telecommunication, consumer electronics, automotive, medical systems and other complex equipment.

The case study results can be summarized as follows:

- The ASD:Suite modelled the behaviour and hardware interfaces of the controller, verified the models and generated C# code
- Using a rigorous evaluation method, Sioux found the ASD:Suite to perform strongly in terms of Productivity and Quality and be better than the average score in six of their ten evaluation categories
- Sioux established that with the productivity and defect free software achievable with the ASD:Suite, they could produce higher quality software in a shorter timescale and simplify future maintenance

The project showed that Verum's ASD:Suite brings valuable benefits to software houses. They can deliver projects on time with fewer resources, enabling them to improve margins, or they can accelerate development to be more competitive and increase revenue. The quality of the deliverable means the company's reputation is enhanced.

Following the success of this project, Sioux is planning to use the ASD:Suite for projects in its Development Centre. The company is also recommending the use of the ASD:Suite to its own customers, offering consultancy in how to use ASD most effectively.

"The ASD:Suite helps you to think before you start implementing, so you get a better architecture and avoid unnecessary complexity. It also finds bugs that otherwise would only be discovered during the integration phase. The end result is a higher quality and more easily maintained product."

Ger Schoeber, Executive Consultant at Sioux

¹ An introduction to ASD, <http://www.verum.com/resources/papers.html>

Following the success of this project, Sioux is planning to apply ASD to projects in its Development Centre. The company is also recommending the use of the ASD:Suite to its own customers

1 Introduction

Sioux supplies software development services to telecommunication, consumer electronics, automotive, medical systems and other complex equipment builders including ASML, Assembleon, FEI Company, NXP Semiconductors, Océ, Philips, Sony and Thomson. It employs 170 engineers based in The Netherlands, Belgium and Russia.

The company continually searches for ways to improve its processes; it believed that Verum's ASD:Suite might help in two key areas:

1. By making development teams more productive, enabling Sioux to reduce production costs
2. By helping the company manage the ever increasing complexity of its software projects, especially during the maintenance phase

Sioux Embedded Systems ran an evaluation project to test whether the ASD:Suite could deliver against these two objectives.

The company selected a central component of a vending machine for the trial. This component was particularly well suited owing to its behavioural aspects, and the fact that in the vending machine the component interfaces to a wide range of hardware units.

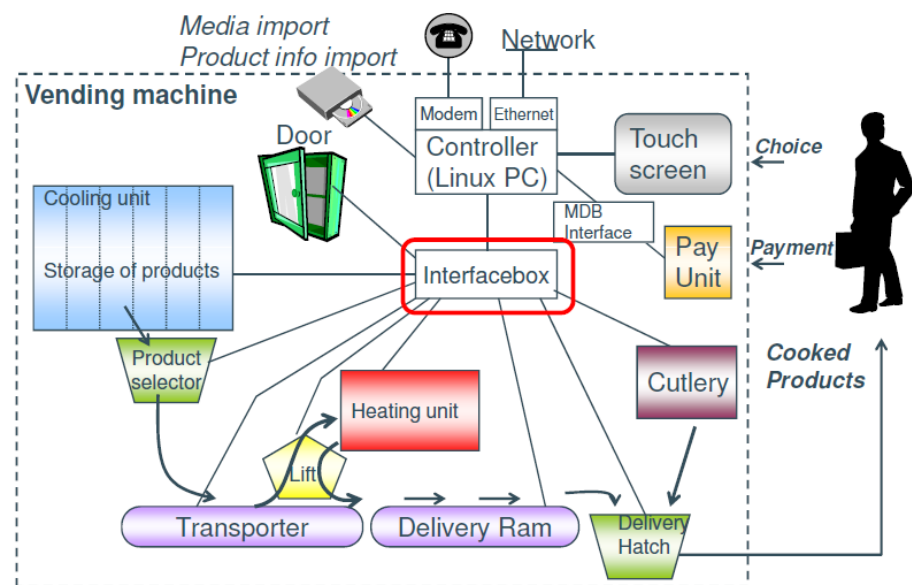
This paper presents an overview of how Verum's ASD:Suite was applied to the re-development of software for the IBoxRequestController, a component of a Fri-Jado hot food vending machine, and Sioux's evaluation of the project.

Sioux had detailed metrics that were collected when the embedded software in the component was initially developed. These were compared with measurements taken during the ASD:Suite evaluation project.

Many of Sioux's projects entail working on embedded systems that interface to physical items like hardware controls. Sioux was confident that if the ASD:Suite proved helpful in the evaluation project, then benefits of a similar type and magnitude could be gained in other projects.

2 Background to the project

The Fri-Jado vending machine contains several hardware components as depicted below. A cooling unit is used to store the products, prior to selection by the customer at a touch screen. After payment, most products need to be heated before delivery. A transporter unit transfers these products via a lift to the heating unit. Here they are heated according to the correct recipe, before being transferred via the lift to the delivery ram for transportation to the delivery hatch. Finally, appropriate cutlery is dispensed along with the food in the delivery hatch. The Interfacebox controls the working of these hardware units under the command of the



IBoxRequestController software.

The controller is a known product with existing metrics, captured when the embedded software was first developed by Sioux, enabling direct comparisons to be made with ASD. In the original development, the controller software consisted of 3423 lines of Java code, generated from 450 hours of design and coding.

Sioux wished to establish whether using the ASD:Suite for this development would:

- Reduce costs, by lowering the number of hours required for design and code generation
- Produce code that was more clearly documented, with fewer defects, and which hence would be easier to maintain

The controller is a known product with existing metrics, captured when the embedded software was first developed by Sioux, enabling direct comparisons to be made with ASD.

3 Method

Sioux wanted to use the evaluation project to learn how to utilise the ASD:Suite effectively. Hence, the company allocated a senior software engineer to the project full time and a senior software architect for 20% of the time. Assistance from Verum was provided as and when required.

The project lasted two months and covered the following key activities:

- Extraction of requirements from existing documentation and the original implementation
- Modelling of the behaviour and interfaces of the controller
- Verification of the models using the ASD:Suite ModelChecker
- C# code generation using the ASD:Suite CodeGenerator

Metrics were carefully collected, so that they could be compared with measurements taken during the original development.

Metrics were collected, so that they could be compared with measurements taken during the original development. Sioux was careful to distinguish learning time from time spent actually using the ASD:Suite for design and development.

3.1 Productivity metrics

The table below shows how Sioux used the data gathered to calculate productivity, in terms of lines of code (LOC) generated per hour.

Productivity Metrics			
	Conventional	ASD, including learning	ASD, excluding learning
C# LOC generated		2889	2889
Java LOC written	3423		
Hours spent	450	456	316
Improvement over conventional		-1%	+30%
Productivity during design and coding	7,6	6,3	9,1
Improvement over conventional		-16%	+20%

Notice that no integration or system testing was carried out during the evaluation project. Other studies have shown that productivity improvements during design and coding using Verum's ASD:Suite are enhanced still further during integration and system testing, so that overall productivity increases of 30% and more can readily be achieved.

3.2 Evaluation

The Sioux team were rigorous in their evaluation of the ASD:Suite. They produced a 20-page internal report to document the evaluation results and best practices that were captured during the project.

The ASD:Suite ModelBuilder was employed to create Interface Models and Design Models, generate source code and launch the ModelChecker. The report shows examples of ModelChecker output and how this output can be used to correct and validate the Interface and Design Models.

Project goals covered ten key areas including:

- Economics
- Maintainability and Scalability
- Quality and Productivity Gains
- Ease of Use

In the evaluation, Sioux rated the ASD:Suite in each of these key areas.

3.3 Key benefits

The Sioux team reported that ASD models and generated code enable rapid prototyping early in the project. A main strength of the ModelChecker is that it addresses deadlocks, livelocks and race conditions, again at an early stage. Discovering these without the ASD:Suite is a difficult and time consuming task for software engineers, and there is always the risk that some are not discovered before product release. For these and other reasons, the productivity of each engineer increased.

Additionally, Sioux found that the ASD:Suite encourages the software engineer to simplify the model and keep it simple, and to stabilize the logical operational flow before descending to physical implementations. The generated code is well structured and guaranteed to be defect free, which improves the quality and maintainability of the delivered product.

The ASD:Suite increases the productivity of software engineers, reducing development time, and generates well structured, defect free code, thereby improving the quality and maintainability of the delivered product.

4 Results

Applying Sioux's rigorous evaluation method, the project team found the ASD:Suite to be better than the average score in six out of ten areas.

It was the Productivity and Quality gains which really convinced the Sioux team that the ASD:Suite could be valuable to the company ...

Evaluation Summary		
Item	Value	Weight factor
Economics	2,7	3
Productivity	4,0	3
Quality	4,2	3
Ease of Use	2,2	3
Security	4,0	2
Maintainability	2.7	2
Scalability	3,7	2
Scope of Method	3,5	1
State of Tooling	3,0	1
Learning Curve	2,3	1

Under the Economics item, Sioux calculated a saving of 16% in the costs of design and code generation. Other studies have shown that reductions in the Integration and System Testing time can raise overall savings using the ASD:Suite to around 30% of development costs.

However, it was the Productivity and Quality gains which really convinced the Sioux team that the ASD:Suite could be valuable to the company, especially if it was applied to software projects for embedded systems that have interfaces to physical aspects of the surrounding environment.

The table below summarises the cost comparison between the original software development and that using the ASD:Suite.

Metrics			
	Original	ASD	
		Learning included	Learning excluded
Effort spent	450 hrs	456 hrs	316 hrs
Lines of Code	3423 Java	2889 C#	
Cost			
Hour costs	€ 31.500	€ 31.920	€ 22.120
ASD costs	-	€4.400	€4.400
Total costs	€31.500	€36.320	€26.520

5 Conclusions

Sioux has been developing products at its own Development Centre since 1997. At this centre of expertise the company works on new technologies for its customers to incorporate into their own products. Sioux management accepted the recommendations of the evaluation team, and the company is currently determining which projects in the Development Centre will gain the most benefit from using the ASD:Suite.

Sioux believes its customers can directly benefit from ASD. Having learned how to use the ASD:Suite, Sioux is now planning to offer consultancy to its customers in how to apply ASD most effectively.

This evaluation project has clearly shown that ASD modelling helps in designing, verifying and generating code for embedded software. The significant increase in productivity achieved enables development to be completed more quickly, with other studies complementing the Sioux findings to show that Verum's ASD:Suite can reduce software development time and costs by around 30%. The defect free software generated by the ASD:Suite contributes to the improved quality and maintainability of the delivered product.

These results are highly valuable for software houses. With the ASD:Suite they can deliver projects on time with fewer resources, improving margins. Alternatively, they can speed up development to be more competitive and take on more work. Since the quality and maintainability of the delivered product are improved, the company's reputation amongst its customer base as a supplier of software services is enhanced.

The results are highly valuable for software houses. With the ASD:Suite they can deliver projects on time with fewer resources. Alternatively, they can speed up development and take on more work.

6 About the authors

Leon Bouwmeester is an ASD Consultant at Verum Consultants BV.

Arjen Klomp is Commercial Manager at Verum Consultants BV.

Acknowledgements

We are grateful to Sioux for allowing us to present this case and for their effort and cooperation in applying ASD to the re-development of software for the IBoxRequestController, a hardware controller located in the Interfacebox of a Fri-Jado hot food vending machine.