



AADL and Télécom Paris Tech

featuring Peter Feiler and Etienne Border interviewed by Suzanne Miller

Suzanne Miller: Welcome to the SEI podcast series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is [Suzanne Miller](#). I am a principal researcher here at the SEI. And today I am pleased to introduce you to my friend and colleague, [Peter Feiler](#), who is a technical lead and author of the Society for Automotive Engineering's [Architecture Analysis and Design Language, known as AADL](#). Peter's research includes dependable real-time systems, architectural languages for embedded systems, and predictable system analysis and engineering. In 2009, he received the Carnegie Science Award for information technology for his work with AADL. The AADL standards committee is meeting in Pittsburgh this week with members from throughout the globe to discuss evolving elements of the standard and to work together on action items from prior standards meetings. Peter, I understand you've brought Etienne Borde from Paris to see us today and talk with his about his involvement in AADL. Would you please introduce him a little bit?

Peter Feiler: Sure, yes. Etienne Borde is from [Télécom ParisTech](#). That's a technical university in Paris. He will tell you a little more about the kind of work they're doing. But, he and I got to know each other through the committee because that university has been actively working with the committee for a while. And in that context, we also became colleagues in working together. He has spent time even at the SEI to do some of the research work.

Suzanne: To work more closely with you?

Peter: Exactly.

Suzanne: So, Etienne, I'd like to know a little bit about what you see as the reason that Télécom Paris is interested in AADL. Why is it important to you? Then, what have you been doing with the committee with the AADL standard and its annexes?



Etienne Borde: I would say first that the reason why Télécom ParisTech is interested in AADL is because the whole embedded systems industry is growing up quite fast, especially in Europe or in France. We are really interested in this domain. Télécom ParisTech is, of course, a research and a teaching institute. So, we have to produce research results and teach students how to make better embedded systems.

Suzanne: So, you use AADL as a teaching tool, not just as a research tool?

Etienne: Yes, we use it as a teaching tool and as a research tool, and both parts are a little bit challenging. The research, of course, because we have to face new problems. The teaching because AADL is typically the kind of language that you can use for very advanced software engineering methods. We tried to teach it to students that physically did very little software in the labs. So, there is this kind of gap between what they're able to understand about software engineering challenges and what AADL is meant to answer. So, that's also funny.

Suzanne: Well, and it's a challenge because these are systems engineers that have to work with the software. So we need them to understand more about how the software affects their decisions, or how their decisions affect how the software will perform. So AADL can help with that if we can teach them how to make good use of it.

Peter: Exactly. On the software side, it would be students who learn Java because they wanted to write some scripts for some web pages or some basic simple programs.

Suzanne: That's a little bit different than writing a navigation system for an airplane. I hope. It needs to be different.

Etienne: This is typically the reason why we are interested in AADL as well. It's because it's a very good language for us, in our opinion, to tackle the issues that you have in safety-critical, embedded systems. So, those for which you need to have very strong guarantees in the behavior of your software applications.

Suzanne: And, you want it to be provable.

Etienne: You want them to be provable. You want those systems to react in a predictable way in case there is something that happens while the user is using it.

Suzanne: So, what parts of the standard or the annexes have you worked on?

Etienne: We're mainly using AADL in our research work to make the prediction of the software applications or the configuration of the operating systems that will host the software applications.



The operating systems in safety-critical, embedded systems have very different characteristics than in standard computer systems. Of course, you can't accept that your operating system fails the same way that your home operating system could fail.

Suzanne: You can't have a blue screen of death. That's not allowed.

Etienne: You cannot have blue screen. You can't have delays of tasks. You need to have specific operating systems. Those are quite difficult to configure. You have to be careful in the configuration process of those and the type of guarantees that you can manage by this configuration process.

Peter: So, what makes that work interesting is the following: So far, we have talked primarily about analyzing the architecture upfront to find problems. What they are doing now is—given that we have a model that has been analyzed and shown that the model itself is correct—we are now automating the process of generating some of the glue code from a correct model rather than from a piece of paper that says, “And, here's what you're supposed to do.”

Suzanne: So, that removes sources of error?

Peter: Yes. The next phase of the process is we are basically pushing it down into the whole development process that we can do: end-to-end-verification, early on requirements, and high-level architecture, find things early. As we refine our design, still do it model-based. As we go into code, drive it down.

Suzanne: So, this connects with a lot of the model-based software engineering that the EU [European Union] has been researching over the last several years. I was involved with some of those things back in the '90s. It goes back a ways. So, okay, I see that connection.

Etienne: We are typically—in terms of model-driven engineering—doing model transformations. As part of the code-generation process, we produce the glue code at the end. So, we try to have AADL in the picture at different steps in the development process or in the code-generation process, to have, first, an input model on which you can make some analogies and then automate the prediction of the new AADL model, in which you can, again, do some other analyses, but in a more precise...

Suzanne: But it's more specific.

Etienne: It's more specific.

Suzanne: More precise.

Etienne: Exactly.



Suzanne: Because one of the really amazing features about AADL as a language is it will accept inputs at different levels of precision.

Peter: Exactly.

Suzanne: So, early on, you don't have a lot of precision in your parameters. But, then, you're able now to do a follow-on model that is more precise and gives you more details about performance.

Etienne: Yes, that's exactly correct.

Peter: The other part is that the individual pieces of application code may be either written in [Simulink](#) or in whatever. Again, AADL provides the glue.

One of the interesting ways of making use of the code-generation capability that they have is, I may define my architecture model as a task architecture. Analytically, look at what my end-to-end latency and response time is. But, normally, I wouldn't know until the software finally is integrated and runs, whether the timing will...

Suzanne: Will actually play out, yes.

Peter: In this case we can actually already generate the complete run-time system and, for the actual application code, just put dummy code in there. No piece of application code has been written, but we have a fully running system, so we can...

Suzanne: The platform is done.

Peter: We can see the timing characteristics and can measure, *How long does it take if a sensor would produce data? You do all your computations before it goes out.* Not just based on an analytical model, but based on actually running software. It can do that much earlier in the process as well.

Etienne: This is interesting because often industrial partners that we work with have a strong intellectual property or a strong-added value in the functional part of their software. So, they don't want to say much about how this works. But, if they can have some knowledge about how the infrastructure will work when they will embed their functional code inside it, it's also something interesting for them. To have this clear separation between how you configure the platform, and, then, they can work on their own function.

Suzanne: On their added value.

Etienne: Yes.

Suzanne: OK. Well, that's wonderful, Etienne.



Suzanne: Well, I'm very glad that you've been able to join us this week in Pittsburgh. This is a return trip for you since you spent some time with us before.

Etienne: It's a very nice city to visit and to live in.

Suzanne: I agree. I agree. Peter, thank you for bringing Etienne to us today. Etienne, good luck with your code generation and model validation, all of your work. And thank you very much for supporting AADL.

Etienne: Thank you. Thank you so much.

Suzanne: If you, our listeners, would like more information about AADL and the work of the standards committee, please visit the AADL wiki site at <http://www.aadl.info/aadl/currentsite/>. That's all one word.

Peter: Or, in short, www.aadl.info.

Suzanne: This podcast is available on the SEI website at [SEI.cmu.edu/podcasts](http://sei.cmu.edu/podcasts), and on [Carnegie Mellon University's iTunes U site](#). As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu. Thank you.