Open Systems Architectures: When & Where to Be Closed

featuring Don Firesmith as 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 sponsored by the U.S. Department of Defense. 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. Today, I am very pleased to welcome my colleague and friend, Don Firesmith. He is going to talk to us today about open systems architectures (OSA). Before that, let me tell you a little bit about him.

He is a principal engineer here at the SEI, and he supports the U.S. Department of Defense and other government offices in the acquisition of software-intensive systems. He provides practical guidance with regard to requirements engineering, software and system architectures, and you did not say testing on here Don. You have given advice about testing too, but that is a different podcast. Today we are going to talk about architectures. And, welcome.

Don Firesmith: Glad to be here.

Suzanne: Why are open systems architectures a priority for the DoD right now? They have sort of been in vogue, out of vogue, and now they are back again. So, tell us: why now?

Don: Let me start first by clarifying what open system architectures are, because everyone might not know the same thing. There are actually two kinds of OSA’s. There is the open system architecture, which basically means that your system architecture is modular. These key modules, these key system components, have interfaces that conform to open interface standards. That is basically an architect’s view of what OSA means.

There is also an open system approach, which is more of a business or management kind of thing that your typical program manager or project manager is interested in.

Suzanne: That is about the governance of all of these components and making sure the standards line up and all those kinds of things.

Don: To get the benefits of OSA, you really need both the technical side of the architecture and the business side of the open system approach. Now, why it is important is an open system architecture provides lots of benefits or at least makes it much, much easier to get lots of benefits. Probably the most important one is increased competition.

If you have these modular components that can be separately procurable, and you can then put that component out for bid with multiple contractors, this helps you avoid vendor lock, which is a very major problem in the defense world because once a major defense contractor has a system, typically they keep it for a very long time. It improves maintainability. By keeping things modular, you localize the impact of changes, so you are able to make changes here without big influences in other places.

Suzanne: So, that reduces the avoidable complexity.

Don: Yes, and it certainly reduces lifecycle costs. There are times in fact when this can help you increase speed to the warfighter for a defense system. If you have to support small businesses, you can allow smaller businesses to compete for these modular components.

Suzanne: Because they don’t have to be able to produce the entire system. They can just produce the thing that is part of their niche that they provide.

Don: That is very good, because very often it is the small, agile companies who really shine in certain very restricted areas but are not able to do an entire system. Then the last example is if you want to increase foreign military cells, you have to get rid of the components that would be classified *U.S. only*. If that component you have to get rid of is one of these modular components with the nice interfaces, you can yank it out. You can put another one in that does not have these problems.

Suzanne: Forbidden aspects. This is a good thing. All these things sound really good. There have got to be some challenges, because if there were not any challenges everybody would be doing it already, and we would not be talking about it here.

Don: Well, there are challenges out there, some for the program office and some for the contractor. The program office is typically responsible for the open system approach. This is a new way for them to do business. It has impacts on contracts and things like that. It is not always easy for people who are used to doing things the same way they have done in the past … to make those changes.

When it comes to the open system architecture, that is an interesting issue because that could be the responsibility of the contractor, if the contractor comes up with it. You see that quite often. We are seeing more and more that the government might be the system integrator or the main developer. Often it would be the government who would pick these key components that they want to be separately procurable.

Suzanne: And what standards you want them to be conforming to.

Don: Exactly, so that is one of the issues that has to be dealt with is who does what and it’s new things for people to do.

Suzanne: Historically, some of our program offices that have become accustomed to outsourcing essentially all the architectural aspects, they do not necessarily still have the staff and the competencies within their workforce to be able to be comfortable doing this. So, there is a transition period if you want to move in that direction. There is work that has to be done in terms of workforce development, on the government side, to bring people up to speed on what are all the issues and what are the technical challenges different systems have.

Don: That is one of the reasons why the government often has support contractors that provide their technical expertise in these areas.

Another interesting issue is the program offices tend to think in terms of systems engineering because they are acquiring a system made of subsystems. Well, very often when we talk about an open system architecture, we are talking about that open software architecture as well as an open system architecture. So, the people might, you know, talk past each other. Another thing is contractors naturally like to produce proprietary systems and get that vendor lock-in.

Suzanne: They want to own the intellectual property that goes along with it.

Don: Exactly, so there is this natural tension between the government, which wants to own the intellectual property that owns basically the architecture, and the contractor who wants to own the architecture.

Suzanne: Although, from the contractor viewpoint, if they do have business in other than military, like dual-use kinds of things, they get the advantage of open systems architecture in terms of things, like you said, the foreign military sales. They can actually offer things to customers that they might not otherwise be able to if they adopt this approach.

Don: Absolutely, because in this case they are the owners, and they get the benefit from it as well.

Another thing is, there is still confusion after so many years as to what exactly OSA means, what is improved, what is not.

Suzanne: And you and I are part of a working group at the SEI where probably the first three meetings that we had were just all about that, *What is in and what is out of that definition?* So, that reflects the larger community as also having that problem.

Don: Another interesting problem is that the contracts usually say something to the effect of, *Thou shalt have an open system architecture*. Well, the problem is that open systems architecture, open systems is a matter of degree. There [are] really no systems that are 100 percent open or a 100 percent closed. Now, the question is *To what degree and in what areas should the system be open?* and *What areas and to what degree does it not matter?*

Suzanne: Sometimes I find when you see those sweeping statements, whether it is in a contract or other settings, it is really because the people that are asking for it know they need this, whatever *this* is, open systems architecture. But, they do not really have their own understanding of what they want ironed out enough to be able to give more clarity, so you have got to have a dialogue with whoever you are contracting with to make sure that you do get what you intend.

Don: Exactly, and in fact, it is not just the dialogue but also the contractual requirements. For example, I recently developed a set of reusable requirements for open system architectures for the Air Force, which started out with *What is an open system architecture*? and then derived more detailed requirements under that, and then more detailed requirements under that, so that when you got down to the bottom level requirements, you were specific about what you meant.

Suzanne: But that is not as typical today.

Don: No, it is not typically. You just see, *You shall have an open system architecture*. And that then raises the interesting issue of *If the requirement is not that specific, how do you assess whether or not the contractor actually does have an open system architecture or whether or not the architecture is open in the right ways to the right degree?*

Suzanne: To meet the business goal. This is where the business and technical interact is, when I am asking for open system architecture, I need to understand what my business reason is for it so that I put the right governance, the right evaluation-kinds-of mechanisms in place.

We are seeing this trend, but whenever I see a trend like this, no matter what it is, I always ask the question—and I know you do, too–*Just because we can does it always mean we should*? Are there times when you would think that OSA is not an appropriate thing to bring into a complex system development?

Don: Right. As I said before, no real system tends to be a 100 percent open or a 100 percent closed anyway. We do have to address the fact or the question of where do we want it to be open and that goes back, as you said, to why?

One thing to think about is openness is merely one of the architectural characteristics or attributes that you might want to have in your system. Naturally, you might want to have high performance, safe, secure, all of the *-ilities*, all of the quality factors. These are not always…

Suzanne: They are not always compatible.

Don: They are not always compatible. They do not always support themselves and each other. A good example of this is performance. Regardless of what quality factor it is, almost always the more of that, the harder it is…

Suzanne: The more security, the harder it is to get fast performance. As end users we all deal with that.

Don: Another thing to look at here is one of the main reasons for doing open system architectures is to promote competition. What if you are in an area where there is no competition? Where basically one…

Suzanne: It is such a specialized domain.

Don: Yes, one specialized domain, specialized contractor basically has it sewn up. Now that does not mean that you don't necessarily want it to be open so that in the future other businesses can compete for at least part of it. But, if there is only one main competitor, the current reasons for making it open are less important than they would be otherwise.

Sometimes, the interface standards that these interfaces have to conform to, they might be immature. It might be an area where there is no open interface standard or the open interface standard that does exist may not provide adequate security, for example. Once again you are getting into the competition.

Suzanne: One thing to think about, I mean you said this almost explicitly, is that *openness* you can think of as a quality attribute that you trade off along with other quality attributes. So, openness in comparison to security, openness in comparison to performance; this is one of the things that needs to come into that discussion that may not have been part of it before.

Don: That is one of the reasons why it is not a good idea in general just to have a requirement that you shall have an open system architecture because that is too simplistic. In fact, the Department of Defense policies, the Better Buying Power and so on, recognize this fact, and they introduced the concept of a key interface.

Well basically, certain components, certain modules, architectural components are the ones that you might want to be able to swap out or compete and so on. Well, the interfaces for that module are the ones that are key and therefore the ones that you really want to have them conform to an open systems standard. Now, other components may not need to be that modular, or they may have their modularity driven by security or robustness, safety, or whatever.

Suzanne: And the standards that they use for communication may be closed. They may not be themselves open.

Don: Right. Sometimes, for example, you might have a small group of contractors and the government get together in one specific area. They may come up with their own standard way, standard protocols, for example, for that one specialized area. Now these protocols, they are not ISO [International Organization for Standardization] standards or ANSI [American National Standards Institute] standards…

Suzanne: They are local standards. They are not global.

Don: They are very local standards, so they are less open, if you will, than say an ISO standard might be. So…

Suzanne: Within that community, it is a level of openness that may not have been achieved before.

Don: Yes, and in fact it may be the appropriate level of openness in that community.

Suzanne: If I am a DoD program manager, I am a system architect, based on what you have seen in this area, what are some takeaways? What are some things that I should be thinking about?

Don: If you are a government program manager, then probably the most important thing to decide up front is *Who is going to be responsible for engineering the open system architecture*? *Is the government going to come up with the architecture and then mandate that on the contractor? Or, is the government going to leave it up to the contractor to decide the openness*? There's pros and cons with each way, but that is one thing that has to be decided fairly early on.

Another thing is, if you are the program manager, you want to make sure that based on your business goals—based on whether or not you are responsible for the open architecture or the contractor's going to be—you need to write clear requirements at the appropriate level of abstraction.

Now, when it comes to the architects, the first thing, clearly, is to select what these key architectural components are going to be. Then, when you look at their key interfaces, the next thing that you have to decide is *What is going to be the open interface standards that these interfaces need to conform to?*

Then, the other thing that needs to be done here is, just because the government says *You are going to have an open system architecture*, and just because a contractor says that they have an open system architecture, [that] does not necessarily mean that the actual interfaces conform to the interface standards. Part of the definition of openness or open interface architecture is that the compliance of the interfaces to the interface standards has to be verified often by testing but also by code analysis.

I guess the last thing that I would say—is critically important—is there are people out there who are very happy doing things the way they have always done it. That can sometimes be contractors. It can sometimes be government personnel. Nothing in this podcast should be construed as an excuse not to use openness. We want to use openness where it makes sense, where it is appropriate. And, where other quality requirements, other issues take precedence, then it is OK to back off on it.

Suzanne: But as you say, we are not trying to say that the challenges are insurmountable. They are things that we are seeing. I am working with programs, you are working with programs that are doing this to one extent or another in a way that has been very productive for them so challenges come with any kind of change.

Don: That is why I said there is all of these benefits. It makes sense to apply the right amount of openness to achieve the benefits that makes sense for your application.

Suzanne: I know that this is an area that the SEI and you are going to continue to work on. I am in that working group, too, so we are going to be keeping going on that. I do want to thank you for coming and joining us today and talking about this, because I think it is something that we are going to see more interest in both the contractor and the government community on. And we want to be productive in the way that we approach this, so thank you very much.

As I mentioned earlier, in addition to his work on software architectures, Don has also authored a series of blog posts on software testing and verification. I am sure there is going to be some future ones on verifying open systems. If you want to see any of these posts and also to get to any of the other work that Don has produced, please visit insights.sei.cmu.edu. Click on the Authors button in the upper left-hand corner and find Don Firesmith, and a world of new things will open up for you.

We will be including links to some of your other work that are mentioned in the podcast in our transcript. And this podcast is then available on the SEI website at sei.cmu.edu/podcasts. It is also available on the Carnegie Mellon’s ITunes U site. If you have any questions, please do not hesitate to contact us at info@sei.cmu.edu. Thank you for listening and for viewing.