PPI SyEN

SYSTEMS ENGINEERING NEWSJOURNAL

EDITION 123 | APR 2023

BREAKING DOWN BARRIERS TO SE
Ignorance, Misunderstanding and Ego

SysML v2 DESCRIBED
Differences between SysML v2 and SysML v1

SYSTEMS ENGINEERING RESOURCES
Improve your SE effectiveness

A PROJECT PERFORMANCE INTERNATIONAL PUBLICATION | PPI-INT.COM

PPI SyEN

EMAIL: PPISyEN@PPI-Int.com

EDITORIAL STAFF

Editor
John Fitch

Editor-in-Chief
Robert Halligan

Managing Editor
René King

PRODUCTION STAFF

Marketing Manager
Benjamin Bryant

Graphic Designer
Matthew Wong

Marketing Coordinator
Rebeca Carneiro

Publishing Assistants
Trudy King
Shalani De Silva

PUBLISHER

PROJECT PERFORMANCE INTERNATIONAL

Project Performance International
2 Parkgate Drive
Ringwood, Vic 3134 Australia
Tel: +61 3 9876 7345

Tel Brazil: +55 12 9 9780 3490
Tel UK: +44 20 3608 6754
Tel USA: +1 888 772 5174
Tel China: +86 188 5117 2867

www.ppi-int.com

PPI SyEN (PPI Systems Engineering Newsjournal) is published monthly.

Archived editions and subscriptions to future editions are available for free at: https://www.ppi-int.com/syen-newsjournal/

WELCOME

Hello dear readers,

I am beaming with delight as I write the welcome words for the April 2023 edition of PPI SyEN. This edition is one of the most exciting editions we'll see this year and I say that confidently even though we're not even halfway through the year. A scan of the titles of this month's two major contributions, firstly by PPI consultant Randall Iliff and secondly by SysML experts Sandy Friedenthal and Ed Seidewitz, will give you insight as to why.

This month we kick off the edition with the general question: If SE is so valuable, why is it not being adopted on a wide scale? SE, or anything like SE under a different name, is adopted by a very small percentage of people as evidenced by the macro and micro human-caused disasters that occur around us.

I'm sure that we each have our own tales within our working experience in which we have experienced frustration in trying to convince a manager to invest time in what we know is a sensible development approach e.g. in understanding the problem before leaping ahead with development. The pain is obvious and the solution is available – how do we get our message to the masses? Randall's article unpacks this topic eloquently and fervently. This is probably going to be a read you will recommend to your colleagues for years to come.

As the Newsjournal progresses, we get into more specific-to-systems engineering content until we reach the acme of the Newsjournal in our feature article by the abovementioned SysML authors. The article highlights the key differences between SysML v1 and SysML v2. We on the PPI SyEN team have been eagerly looking forward to the article and our expectations were exceeded in this articulate and detailed summary of the differences. MBSE champions and consumers, this one is for you.

Apart from these exciting articles, we have news and resources pertinent to SE. In addition, we have a mighty set of conferences and events featured in this month's Newsjournal. From business analysis events, tool vendor conferences, resilience engineering and other domains, there are so many amazing events taking place. We have highlighted thirteen events that I personally would attend if I had the time and capacity to do so. However I am excited to share that I will attend the IISE Annual Conference & Expo in New Orleans in May 2023 alongside this month's Forum author, Randall Iliff. If you'll be attending this event, please reach out on LinkedIn or via email. I would love to sit down and have a coffee with you at this event.

Regards,

René

Managing Editor, PPI SyEN

PPI Systems Engineering Newsjournal (PPI SyEN) seeks:

- To advance the practice and perceived value of systems engineering across a broad range of activities, responsibilities, and job-descriptions
- To influence the field of systems engineering from an independent perspective
- To provide information, tools, techniques, and other value to a wide spectrum of practitioners, from the experienced, to the newcomer, to the curious
- To emphasize that systems engineering exists within the context of (and should be contributory toward) larger social/enterprise systems, not just an end within itself
- To give back to the Systems Engineering community

PPI defines systems engineering as:
an approach to the engineering of systems, based on systems thinking, that aims to transform a need for a solution into an actual solution that meets imperatives and maximizes effectiveness on a whole-of-life basis, in accordance with the values of the stakeholders whom the solution is to serve. Systems engineering embraces both technical and management dimensions of problem definition and problem solving.

PPI SyEN FORUM

Selected correspondence from readers, authors, and contributors

Why Recognition of Profit, not Imposition of Process, Will Ultimately Bring Systems Engineering Into Common Usage

by Randall C. Iliff (PPI Course Presenter and Principal Consultant)

Email: riliff@ppi-int.com

Abstract

Within INCOSE we tend to focus our energy primarily around the tools, methods and practices that define our elegant engineering art. That emphasis on the "elements" and "connections" within Systems Engineering (SE) makes perfect sense for those who operate primarily inside the profession but can easily hide a much larger awareness of the "emergent properties" that even the most rudimentary *functioning system of SE* offers the world.

The logic (and to an increasing extent the model-based methods) embedded in SE are well matched to the dynamic system called reality around us, and thus enable the possibility of routinely manageable *invention-on-demand*. The logic remains valid from the simplest to most complex systems and is arguably most valuable not at its extreme upper limit but rather in aggregate at the threshold of enabling a result that was not previously obtainable for a given individual or team. Thanks to almost limitless technology, our ability to convert imagination into reality has never been easier, nor available to so many people.

That combination of attributes, in the hands of visionary leadership, is a virtual license to print money. Whether you lead the product development team of a major corporation or are an average individual with a dream anywhere on the planet, SE has something powerful to offer you. The world is filled with marginal systems just waiting for something a little better to arrive.

Given that every market force would favor adopting SE as quickly as possible, why do so many leaders stubbornly ignore this opportunity? The scrutiny placed on financial measures of success, particularly the ability to generate profit, would certainly seem to favor giving SE a try.

Fads get funded and followed with religious zeal – until the next one. Since fads are at best incomplete and at worst incompetent, they inevitably solve some problems but introduce others – which in due time become what the next fad fixes. Tooling for innovation is at least as valuable as tooling for production - why not just invest in an engineering competency that is fundamentally aligned with

reality and then continuously improve your abilities?

At the 2022 INCOSE International Symposium (IS) it was my privilege to share a presentation titled "What Force is More Powerful than Profit" exploring this fascinating and perhaps existential question for SE. This article builds on that foundation, starting with the reality of SE's value, then identifying key barriers and sharing ideas for overcoming them, and finally concludes with a request to help the SE community at large by participating in a short survey of the barriers that you encounter in your unique environment.

We'll also host a conversation on LinkedIn and welcome your thoughts and experiences. The findings from the poll, along with key discussion insights, will be shared in a follow-up article later this year.

Introduction

What are the real Barriers to SE?

The story is not about somehow proving that SE is valuable, but rather exploring why such undeniable value has gone unnoticed by those who otherwise embrace, often irrationally, the slightest promise of advantage in finance, production or business operations. Further, the barriers that prevent awareness have held firm, despite an evolving series of tools, messages, and outreach efforts.

The nature of those barriers has fascinated me for decades.

What I first naively thought of as discrete gaps in understanding, simply needing to be bridged, could often be better visualized as intentional layers of high-voltage insulation. It wasn't just a lack of curiosity about SE, for an important subset of leaders there was purposeful opposition to even consideration of SE.

The underlying reaction was not "let's hear what they have to say" it was "to arms and repel boarders."

Every benefit I offered was met by an equal but opposite rationalization of why they either didn't need that benefit or assurances it was impossible to use SE in their organization / market / region and so on. Mythology is filled with examples of such epic struggles, none seemed a model I wished to emulate, so I simply cut my losses and moved on. The open market offers the luxury to decline impossible clients, but escape is much more difficult when you are chained to your opponent by contract or employment agreement.

These are dynamic barriers, triggered and sustained by system forces within the world of the decision makers we seek to reach. Any rational message of benefit is doomed when interacting with a clearly irrational system of values. A small number of irrational participants can easily make it impossible to reach and help the rational subset of the organization.

The barriers look different in different SE application contexts of course, but the freedom to act based on self-interest in the commercial product development sector offers a great place for us to start. This sector is free to apply or ignore SE, and the choices that they make absent policy or customer direction are quite revealing.

How Our Messaging Impacts Barriers

As a Senior Systems Engineer, and later a Vice President, at an award-winning product development firm called "bb7," I used SE constantly, but invisibly, not just as an engineering tool, but to also define market opportunities, capture strong IP, craft proposal strategy, establish effective teams and organizational structures, and essentially look into the future in ways our clients and competitors simply could not.

I made the "beneficial emergent properties of applying SE" visible in the primary value structure that our clients actually cared about – profit – and downplayed our activities as "common sense and

experience." The reason was simple. When phrased as "common sense and experience" they felt included and knew how to take credit internally, whereas "SE" was something scary to avoid since it might point to us as the savior and them as the problem.

You could brand what we did "Aardvark" and it would have turned out exactly the same. The logic, not the nomenclature, determined the results. We used SE to engineer not only products, but entire market systems on behalf of our clients, and in turn made many of them heroes within their own organization.

None of that would have been possible had I insisted on calling what we did Systems Engineering instead of just doing it and handing them the result. Instead of SE, it was Marketing that got credit in many organizations, New Product Development in others, even Quality and Project Management were selectively praised for how well "that one went compared to how things usually go around here."

The challenge of willful ignorance

Only the tiniest handful of clients wanted to know how such a small company could efficiently conduct hundreds of development projects a year across a wide variety of industries and product types – and somehow consistently deliver results their vaunted corporate teams had not even imagined possible. One Golden Egg in hand, they went off happily down their path to market.

The nature of that willful ignorance has fascinated me, in large part because of the enormous opportunity cost that arises. The answer turned out to be very surprising.

When I looked at a client's market, through the eyes of a confident SE practitioner operating under deep cover, it was always possible to see a direct path to a system of evolved product solutions, not just the launch product they had in hand at the moment. (A background in SE and life cycle cost estimating gives one a very good framework for understanding the full magnitude of loss that passing up a dominant market entrance represents.)

When they looked at their market, through the lens of an organization that consistently failed on 80-90% of the development efforts they launched, the idea of having one "safe" project in hand was infinitely more appealing than the idea of gambling against bad odds for more.

What I saw as a license to print money and dominate market share looked to them like a solo game of Russian Roulette.

Further, a surprising percentage of leaders had gained their roles based on a single good idea that somehow made it to market – like winning a lottery ticket – and were terrified of backing something that might diminish that (perhaps undeserved) legacy. Once again, the ability to imagine making things better as a result of applying SE was absent, instead they saw it as us getting lucky and them being able to benefit from the happy accident.

Decades before bb7, I worked at Martin-Marietta Denver Aerospace where I once made the observation during lunch that "Gee, Martin-Marietta Orlando is really good at production, Denver Aero is really good at development, maybe we should figure out some way to mix the best of the two." The reaction to my thoughtful gift of opportunity was one best described as sheer horror – "Don't even think about it, we'd kill both organizations within a year!"

Looking through the lens of possibility, we were passing up a logical improvement to overall enterprise effectiveness / efficiency. Through the lens of painful experience, it was far easier to imagine how things could end up infinitely worse. The conclusion based on lack of confidence was that we would endanger the entire organization rather than help it in any way.

Opportunity cost is only a motivator if you see *opportunity* as something that is likely to go well versus randomly blow up in your face. Regardless of the market, I found that an equilibrium between

opportunity and risk had been established, with the fulcrum of that balance-beam being the degree of confidence that things can actually get better.

In less technical terms, our market doesn't trust *us*, because they don't trust *themselves*.

The Value is Real, Undeniable, and Extraordinary

Humans exist as biological systems, live in a complex system, depend on complex systems for every aspect of our ongoing existence, and thrive or suffer in our environment based on system relationships. These relationships are as invisible as radio waves to the vast majority of people living on the planet – but they are everywhere, all the time, unseen but essential, and impossible to prove without instrumentation.

The wisdom of acquiring some capacity to understand and manage system relationships becomes self-evident the moment you first glimpse and accept the overwhelmingly system-dominated nature of reality. Before that moment you can make all the scary stuff go away simply by not looking at it; after, you'll never again feel completely comfortable.

Like acquiring object permanence as a child, the first true discovery of system relationships fundamentally and irreversibly transforms our relationship with the world around us. Until you believe in radio waves they aren't real - to you – and your cell phone is magic.

That distinction reveals SE's critical dependence on some level of system awareness in the mind of the person we seek to engage. Only the system-literate can understand our message of emergent benefit - everyone else just hears cost and complexity.

Our value may be invisible until a developmental threshold is reached in the mind of the person we are seeking to help, but the *existence* of SE's potential value remains as certain and fundamental as electromagnetic energy.

A 200-year explosion of worldwide system complexity has created an immense concentration of opportunity. Civilization is now burdened with a mountain of unmet system needs waiting to be discovered and served through some form of SE. In addition to that "order backlog" there are an almost unimaginable number of emerging opportunities to do systems just a little bit better.

The wonderful thing about mining "system-gold" is that the mountain is not only immense and full of gold; that mountain creates new gold faster than humans could ever hope to mine it. Because that sounds unbelievable, people will continue to choose not to believe it - *until someone discovers they can get rich*.

Because so few people alive today know what "system-gold" looks like, most of that opportunity for improvement is sitting in plain sight and available for average individuals as well as organizations to pursue. The first few prospectors to ride out on a burro and fly back in a private jet will draw attention to the opportunity, and there is an almost unlimited amount of gold in the mountain waiting to be converted into fortunes that create even more attention.

We don't need to sell SE, we just need to poke a few holes in the barriers that prevent recognition of opportunity and let the power of market-driven capitalism do the rest.

So Why Don't Leaders Embrace, or at Least Explore, SE?

Ignorance

The most commonly cited, and certainly valid, barrier to beneficially applying SE is learning that SE even exists and is an option to consider. Awareness is certainly an essential step, but it's relatively easy to trigger awareness in those *who are already system-literate*.

As a community of engineers and seasoned project participants we have a very high percentage of

system-literate members, many of whom are now pretty good at educating each other about the value and potential of SE:

- INCOSE serves as a strong advocate for SE education.
- Universities now offer SE curricula for those just entering the workforce.
- Companies like PPI have delivered training to tens of thousands worldwide, broadening and enriching careers already underway.
- Most SE teams that work in organizations have embedded training and mentoring programs – often motivated and run by senior practitioners.

We love ignorance, because it offers a blank canvas onto which reality can be drawn. Starting from a blank canvas does present challenges though, in particular for those that have not yet reached the maturity point where systems are "real" and must be respected.

Until this essential magic of systems is realized, it will be impossible to communicate the potential of unexpectedly strong benefits. Because only awareness of system benefit creates the motivation to invest time in learning SE, this is a critical development threshold that we must enable in order to have any hope of building professional SE skills.

Seasoned SE practitioners view system-literacy as necessary, but not by itself sufficient, and thus tend to diminish the importance of this epiphany moment. If we think back carefully, though, we can recall the points at which our awareness of the world fundamentally changed. Most of these moments were created not by a learning event, but by participating in life itself.

Where we struggle today against the barrier of ignorance is at this threshold – migrating people from a simple, apparently safe but fundamentally flawed viewpoint, to one that is rich with opportunity but far more demanding and uncomfortable to hold.

We know the prize that awaits, so as "parents of the profession" we encourage others to grow up quickly. Because those we are trying to help know the pleasures of irresponsibility but have not (yet) suffered as a direct result (at least that they will openly acknowledge) we often end up dealing with the equivalent of teenagers who think they are immortal and that risk only applies to other people.

In addition to offering "teenagers" a model that works but requires discipline, we face another challenge in dealing with this group. Any status quo favors some at the expense of others, and not everyone wants people to know that the Earth orbits the Sun - especially those whose income and status depends on claiming the opposite.

We thus face a marginally motivated and highly conflicted general audience, people who don't yet share our understanding of systems, and it should be no surprise we have been far less effective here than with those that are "like us" in fundamental ways. Our current applications for professional learning run just fine, but only on a system-literate operating system, and those compatible OS represent at most a percent or two of total market share.

It is this threshold, reducing ignorance surrounding the very existence of systems, that has the potential to enlarge our community from tens of thousands to tens of millions.

We are focused on creating a small pool of very skilled SE practitioners. That is a noble goal, but unless that core is eventually connected to the larger world by people with awareness of value it will come to very little. That next layer of enabling connections has very different learning needs compared with the core SE community, a much different starting point, and very different motivations regarding personal benefit and risk.

Acquiring object permanence isn't something we go to school to learn – there is no textbook or podcast that "teaches you" that objects remain real even when you aren't looking at them. This an internal transformation – invisible at the level of actual change mechanics – that happens when the

preponderance of evidence and experience ultimately forces us to adopt a different view of our world.

Our best classroom for this aspect of learning is the real world – helping people to interpret their own rich life experience in system terms, and thus unlocking the transformation to system-literacy. It doesn't matter what system they first see, as long as elements are connected in ways that trigger emergent properties it is as pure a system as any other for learning purposes.

It is important to realize that this is almost the inverse of how we typically teach SE to new users, where our focus on execution details presumes our audience is already system-literate. Those who are system-literate *are* ready thrive, those who *are not* are frustrated and become advocates against, rather than for, our profession.

We also need to discuss the difference between innocent ignorance – simply not yet knowing something – versus willful ignorance where some need is being met only as a result of denying truth. Willful ignorance is a form of ego-driven misunderstanding and not correctable through the methods that work well otherwise.

There is also a major difference between teaching a lot to a few so that they can perform well and correct problems, versus teaching a little to many so that they no longer create an endless pipeline of dysfunction that needs to be cleaned up. We are good at helping people who want to do better, but nowhere near as good at intervening with the people who are currently causing most of the problems.

Phrased in medical terms, we still treat education as a remedy rather than a prophylactic.

Misunderstanding

Misunderstanding is a canvas that already has something on it, but in the wrong places or colors for the image we are trying to create. The war-of-silos in most organizations results in many small canvases rather than one large one, so in addition to misunderstanding of contents we also face misunderstanding of the connections between those misunderstood contents.

Unlike the blank canvas of ignorance, this one has emotional attachment to whatever is drawn upon it. Those markings were often created by, and remain linked to, competing brands of methods and policy. These are in effect territorial maps of what each silo controls (or wishes to control) and function in a loose feudal structure. Each is thus locked into a larger system of misunderstanding, even if they themselves wish to become enlightened.

We seek to disrupt that equilibrium, and thus clear potential for conflict arises.

Misunderstanding creates its own parallel system of vested interests, rationalizations, and self-preservation needs. The more flawed the level of understanding, the greater the arsenal of secondary compensating and defensive mechanisms you are likely to face.

There is also fear that, if your method is as good as you say it is, what they've always done before will suddenly look *really bad*.

How do you explain that all it took to finally improve market share was simply to ask the market what it wanted and make sure your product has those attributes? Likewise, try explaining why you are only now writing down fuzzy goals as clear requirements for the first time, or actually verifying that what you planned took place as intended.

All of these improvement actions will seem absolutely obvious in hindsight. How do you answer the dreaded question "Why didn't *you* think of that?"

Perhaps the biggest reason for the ongoing success of fads is that a new buzzword lets everyone pretend that the underlying principles are new and thus get a pass on having abdicated their responsibility to discover and apply them decades ago. Talking to your stakeholders was a good idea

long before "Voice of the Customer" became a best-seller.

The key is that misunderstanding always comes with a secondary system of rationalizations and enabling infrastructure. Against such carefully prepared defenses, our message is very unlikely to prevail. The secret to defeating any fixed defense is simply to go over or around the barriers and thus make them irrelevant. If you lack that ability, not engaging at all is the smarter choice.

You cannot cure misunderstanding until there is a basis for understanding – it is pointless to argue with a 2-year-old, instead you must wait until a moment arrives *in their world* that you can then *seize upon as an illustration*. We can create environments that increase the odds of such experiences, and help people draw as much from their epiphany as possible, but we do not have the ability to reach into other people's minds and create this transformation.

Ego

We have now moved from a canvas to a mirror. The owner not only considers what they see on it as a work of art, they identify with it personally.

Worse, that mirror reflects only certain wavelengths of light – the only colors that matter are the ones that reflect and then reach the eyes of people they are trying to impress. All other wavelengths are absorbed and produce heat in the form of anger / frustration / resentment.

Ego must be treated as an addiction rather than a learning opportunity.

You will face an entire system of defenses and enablers, and the strongest possible opposing WIIFM – a sense that if you win, they die. An addict's energy is devoted to maintaining and defending the addiction, not recovering from it.

The model we can learn from is one of de-tox and remission versus cure – relapse is easy and common. Until the fear that drives inflated ego is addressed, the best you can hope for is to move the symptoms into less harmful patterns.

It is far easier to substitute another, hopefully less harmful, drug than withdraw them completely. Once again we see the mechanism that supports empty fads over genuine learning and transformation. Each fad feeds the addiction, and by promising that the next "hit" will finally do the trick as they enable migration from one dependency to the next.

By contrast, we are offering a life of discipline and sacrifice. That's hard to pitch until the person you want to help hits rock-bottom. You have undoubtably noticed that people take SE a lot more seriously once something terrible happens, and that those moments represent unique opportunities for our community to be fully valued.

Never waste a good crisis, it can offer an essential moment when ego is less important than survival and change finally becomes possible!

What We Can Do?

Pitch Benefits

No one sells cell phones by requiring their customers to get a degree in RF engineering. Instead, a few people apply that essential skill to create a marketable commodity that is recognizable to billions of buyers. What the phone does for people's lives is valued, but the RF engineering that made it possible is invisible at that level.

SE is not valuable because it is SE or that we like it. SE is valuable because of the beneficial impact on other systems that emerges when using these methods as a system. My experience with bb7 validated the ability to easily sell SE *benefits*, and the pitfalls of describing only the methods and

hoping others are sophisticated enough to make the connection.

There is wisdom in the saying "Once you can tell the difference between good advice and bad advice, you no longer need advice." The takeaway isn't that you'll no longer need advice at some point, it's that during the period when you most need assistance you'll be unable to know whether or not the guidance is sound. It is *easier to trust a story of benefits* than one that talks about only the actions.

Speak Investment

At the top of every commercial sector Vee Diagram you'll find only money. For the most serious investors, making anything other than return on investment is a nuisance.

It doesn't matter what they make and sell, as long as the emergent property of profit meets expectations. Small wonder that financial instruments – the direct creation of money from money – are considered an ideal investment compared with the messy and uncertain business of creating products whose inherent value must be converted into cash.

Designing cell phones, running theme parks, manufacturing cars, providing health care, and countless other businesses all exist at the very top solely as a means by which the desired emergent property of profit can be acquired. Proclaimed loyalty to market, mission, and people vanishes the second that the money machine is perceived to be at risk.

That ruthless survival instinct can be aligned with doing the right thing – but only as long as the doing the right thing makes them more money. The incredible potential benefit of SE in profit terms thus opens the door to the potential benefit of SE for overall societal efficiency.

Target Entrepreneurs

Entrepreneurs are the prospectors in our system gold-rush model. They are willing to look in places others have called worthless, and to be alert to a flash of color others would miss. They may not be geologists, but they know gold only moves downward, and thus they explore uphill until the gold stops. Somewhere in between is a fortune waiting to be claimed.

These people are the decision makers, and even more importantly they are the direct beneficiary of those decisions. Yes, some are infantile tyrants, but at least that concentration of power eliminates the dysfunction of competing political agendas so common in traditional organizations.

Entrepreneurs are willing to fail as a tool for learning – that in itself is a system and emergent property example we can build upon. The idea that I will try something three different ways and fail each time is normally considered a bad thing. Unless you see the overall value arising from experience and iteration, the set of failures and associated emergent property of foundational insight is never recognized. To the narrow, non-systemic view of most corporate measurement systems it simply looks like three strikes and you're out.

Entrepreneurs are great at leveraging the skills of others, in effect focusing on the emergent properties any way they can get them and largely outsourcing the element and connection level detail. They are system integrators at heart, building only connections and remaining largely agnostic to the origin of the elements.

Entrepreneurs also embrace another form of leveraging the skills of others - they shamelessly copy any viable model they can find and then try to out-promote or out-execute everyone else. This makes them ideal for accelerating the discovery of SE as a tool for increasing return on investment, but also as an innovation laboratory for broadening the different ways that the logic of SE can be brought into practice.

About The Survey

All SE's have a common interest in modeling the barriers and devising ways to bridge them. Crowd sourcing awareness is a logical way to build a clearer view of the challenge and help the community respond. Since we've hopefully opened your eyes to types of barriers you didn't see as clearly before, this is an ideal time for you to share that clarity with others.

The first three questions you'll encounter on the survey ask about the three primary barriers we've shared: Ignorance, Misunderstanding, and Ego. You'll be asked how strongly you feel each barrier blocks SE in your environment. We'll capture data using a numeric score of 0-5 and provide an opportunity to share examples that come to mind. This part is pretty easy, and you likely already have examples or opinions in mind by now.

Being systems engineers, we understand that those elements are very unlikely to exist in isolation from each other and are far more likely to act as a *system of barriers*. We therefore also ask about each possible element to element relationship.

Starting with "How strongly does Ignorance drive Misunderstanding?" we ask six additional questions that enable us to gather the information needed to populate a full n-squared diagram of these relationships. This is much harder than simply looking at the obvious elements, but even a modest attempt quickly reveals that the barriers we *see* are actually the emergent property of a dynamic, self-sustaining, *system of barriers* that we face.

Finally, being not just systems engineers, but *seasoned* systems engineers, we'll ask an open-ended question: "What else acts as a barrier to embracing SE?"

These ten questions, three on the obvious defined barriers, six that capture relationships, and one to capture emergent properties, offer an extremely efficient way to better describe a surprisingly complex system.

Here's where you can find the survey: https://www.surveymonkey.com/r/T7WPR2L Don't worry, it's short so it won't take much time, and we promise not to spam you with unwanted messages.

Given that the question – *What keeps people from embracing SE the way we do?* – is critical to everyone engaged in SE, we'd like to enable ongoing conversation on this important topic. For that reason we've also launched a discussion area on LinkedIn.

There we will periodically share results during the 90-day survey period and join in what we hope will be a lively conversation. Even if you don't think you have something to add, stop by and see what others have to say. The more people that participate, the more valuable the discussion will be for the community, but also the greater the personal validation of realizing that *you were right* all along, and that others face the same challenges for the same fundamental reasons.

Conclusion

The world around us is in transition from an era in which managing elements was sufficient, to one where only systems literacy and control will be successful. Like all transitions, this one has links to both the past and future systems, as well as a transient system emerging at the event-horizon where the two fundamentally conflict.

What looks to us like a tool for engineering is certainly that – a superior model for confidently creating and managing systems – and SE would be absolutely worthy of our passion if that were the only prize. Not as obvious, but far more important for everyone on the planet, is that every activity – not just engineering – needs to undergo the same transformation from fantasy to reality-based expression.

Until you understand systems, emergent properties are seen as directly controllable and also the exclusive point where innovation should be applied. Both are profound and enormously self-limiting

errors, not only for individuals but also civilizations:

- Society will continue to fail at providing essential services such as health care, energy policy, sustainable resources - even maintaining basic social order – so long as the importance of managing the elements and connections that give rise to those desired emergent properties goes unrecognized.
- All of us - society's stakeholders - will fail to hold our representatives and suppliers responsible for making necessary changes. Demanding "change" is pointless, we should be demanding competence and responsibility instead.
- We will attack the symptoms rather than the root cause of problems and remain much more likely to make things worse rather than a little better. The time lag between recognition and response guarantees a period of dysfunction, and in many cases that lag is so large that it makes closed-loop control impossible.
- We will remain vulnerable to exploitation by those who specialize in breaking weak systems for personal benefit – from common criminals to so-called world leaders – and powerless to respond. The safeguards and checks on power that our ancestors fought to introduce will be removed, and their wisdom and sacrifice wasted.
- We will propose grand solutions with no concept of what is needed to create and sustain them, nor even how to judge which proposals are "best." A room full of kids arguing about why their crayon drawing of a car is perfect does not produce an actual working car, nor does it guide logical investment in technologies or systems that would eventually make some version of vehicle possible. It instead becomes a popularity contest that penalizes rather than rewards rational thinking.

The irony of being human is that none of those important observations has anywhere near the motivating power that greed does. Everything in the list above requires a system perspective to appreciate, and in many cases requires a degree of individual sacrifice on behalf of the greater good.

Greed is the inverse of that and says that systems matters only in terms of what they can do for me, right now. Greed is scored in power and money, with money being the easiest metric for the insecure to pursue and rank. Greed and ego are tightly intertwined.

Greed is a powerful and often negative force, but it's also a force that is sufficiently predictable that it can be harnessed for doing good.

Greed, like gravity, will be there regardless of organizational structure, political party, or world governance view. Greed is described and discussed as far back as human records go, and every bit of that accumulated insight remains valuable today. And even more practically, *future-greed* is the only available force with sufficient power to potentially overcome current-greed at societal scale.

SE works because it is fundamentally aligned to reality, managing the elements and connections that give rise to the emergent system properties we seek. That in turn, opens up opportunities not otherwise available, not just in our core benefit story of engineering systems more effectively, but in the much more widely recognizable language of easy money that only an idiot would pass up.

Who but systems engineers would recognize that the literal inverse of a system perspective – greed – might ultimately become the driving force unlocking a better and more system-literate world?

About the Author

Mr. Iliff has over 45 years of PM and SE experience on developmental efforts ranging in size from a few thousand to billions of dollars and has built a solid record of disruptive innovation in aerospace, medical, commercial and consumer markets.

Randy is a charter member of INCOSE and has been a steady contributor ever since. He works with Project Performance International as a Course Presenter / Principal Consultant and is also founder of his own consulting company - Eclectic Intellect.

Prior to that he was Vice President at the award-winning product development firm bb7, served as Systems Engineering Manager for IceCube, an Engineering Manager at Motorola Government Electronics Group, a Program Manager / Senior Systems Engineer at Martin Marietta Denver Aerospace, and a junior member of the "skunkworks" Advanced Development Group at McDonnell-Douglas Astronautics.

Mr. Iliff holds a BS in Engineering / Industrial Design from Michigan State University, and an MS in Systems Management, Research and Development from the University of Southern California.

FEEDBACK

Do you have questions, comments, affirmation, or push-back for authors and articles in PPI SyEN?

Are there trends in systems engineering that give you cause for celebration – or for concern?

What subjects, themes, or other content would be of greatest interest to you in future editions?

Tell us about it, at PPISyEN@ppi-int.com

"

A requirement is some characteristic demanded or deemed an imperative, of something by someone.

Robert Halligan

SYSTEMS ENGINEERING NEWS

Recent events and updates in the field of systems engineering

INCOSE Q1 Highlights

The 1Q2023 edition of the INCOSE Members Newsletter included highlights from a variety of initiatives and activity reports for chapters around the world. Here is a small sampling of such activities.

Steve Records Joins INCOSE as Executive Director
INCOSE has created an Executive Director position to ensure the continued growth and development of the organization and to support the fulfilment of the INCOSE vision.
As of 4 April, Steve Records has joined INCOSE as its first Executive Director and taken the mantle to provide thought leadership, to act as INCOSE's spokesperson and representative with other organizations, to advise and support the Board of Directors, and to manage outreach initiatives.

Impact Statement 2023
INCOSE has released the 2023 version of its annual Impact Statement. Download here to review the highlights.

FuSE Initiative Scales Up
The INCOSE Future of Systems Engineering (FuSE) initiative is scaling up to make progress toward the SE Vision 2025. FuSE projects are organized in four streams:

- SE Vision and Roadmaps
- SE Foundations
- SE Methodologies
- SE Application Extensions

Learn more about FuSE. To engage with FuSE, contact: FuSE@incose.net

Chapter Updates
A small sample of the fourth quarter highlights from INCOSE chapters include:

- Canada: Hosting monthly webinars on topics such as requirements engineering, robust system architecture and the Unified Architecture Framework (UAF).
- Japan (JCOSE): Released a Japanese translation of the SE Vision 2035. Hosted a two-day intermediate level SE course in collaboration with the Japanese Society of Automotive Engineers (JSAE).
- France (AFIS): In December, hosted the AFIS Academy-Industry Meeting with 170 participants.
- Israel (INCOSEIL): Hosted the Women Advancing Systems Engineering conference with 200 participants.
- Spain (AEIS): Hosted a Systems of Systems event in February. Is preparing to host EMEA WSEC 2023 in April.
- Sweden (INCOSE Sverige): Preparing to host the Nordic Systems Engineering Tour in May.
- UK: Invites submissions at events@incoseuk.org for ASEC 2023 with the theme of "*Embracing the New Opportunities*". Held a Meet the Author session with Simon Wright,

author of Don't Panic! The Absolute Beginners Guide to Service Systems and Services.

Working Group Updates

Various working groups report their progress, including:

- Automotive WG: The sub-group on the Application of SE, Safety and Cyber Security is focused on work around their IS2023 paper, "*Taming the Automotive Complexity Explosion*". Also explored the Automotive Reference Architecture Model (ARAM) and the Risk Analysis & Assessment Modeling Language (RAAML).
- Healthcare WG: Hosting the 8th Annual Systems Engineering in Healthcare Conference in April.
- Human Systems Integration (HSI) WG: Reported on the highlights of the HSI Worldwide Workshop 2022.
- Requirements WG: Posted pre-IW2023 sessions on the RWG YouTube channel. Reported on ten takeaways from their collaboration with the Systems Security Engineering (SSE) WG.
- MBSE Patterns WG: Reports that the INCOSE Innovation Ecosystem Pattern was adopted as the reference model for the American Institute for Aeronautics and Astronautics (AIAA) Digital Twin Reference Model and Case Studies publication.
- Quality Management Working Group (SEQM): Partnered with the Quality Management Institute to offer QMI's extensive educational course and QM certification to members of the working group with a full scholarship.
- INCOSE IT: Reports that the Yammer collaboration platform will become Viva Engage in 2023.

For details on these items and more topics of interest, download the full INCOSE Q1 2023 Member Newsletter.

NIST Launches New Trustworthy and Responsible AI Resource Center (AIRC)

The U.S. National Institute for Standards and Technology (NIST) has announced the launch of the NIST Trustworthy and Responsible AI Resource Center (AIRC) that will provide foundational content, technical documents, and toolkits to enable responsible use of Artificial Intelligence (AI). The AIRC offers industry, government, and academic stakeholders knowledge of AI standards, measurement methods and metrics, datasets, and other resources.

Resources currently available on the AIRC include:

- AI Risk Management Framework (RMF): Voluntary guidance to improve the ability to incorporate trustworthiness considerations into the design, development, use and evaluation of AI products, services and systems.
- AI RMF Playbook: Companion resource for the AI RMF that includes suggested actions, references, and documentation guidance to achieve outcomes for the four AI RMF functions.
- AI Glossary: Promotes a shared understanding and improved communication in trustworthy and responsible AI.
- AI RMF Roadmap: Identifies key activities for advancing the AI RMF that could be carried out by NIST in collaboration with private and public sector organizations – or by those organizations independently.

- Introduction to the NIST AI RMF 1.0 – Explainer video.

Engage with NIST through the AI Visiting Fellows.

A Surge of New Documents Added to the Systems Engineering Goldmine

Last month has seen a surge of approximately fifty new documents added to PPI's Systems Engineering Goldmine (SEG). For convenience, SyEN has grouped these new resources into three sets based on their content and sources:

- Software Architecture, part of a 2008 lecture series published by John Wiley & Sons.
- Capability Maturity Model Integrated (CMMI) documents.
- Other resources, too diverse to categorize.

Software Architecture Documents

Let's begin with lectures on Software Architecture:

Visualising Software Architecture
The objective of this software architecture lecture includes, what is visualisation, differences between modelling and visualisation, evaluation. (Source: John Wiley & Sons)

Visualising Software Architectures, Part 2
Concrete examples of evaluation of a diverse array of visualisations. (Source: John Wiley & Sons)

Analysis of Software Architecture, Lecture 13
Lecture covers architectural analysis and models, concerns, and consistency goals. (Source: John Wiley & Sons)

Implementing Architecture, Lecture 15
This lecture covers topics on concepts, architecture implementation, evaluating frameworks, relationships between middleware, frameworks, component models. (Source: John Wiley & Sons)

Implementation Techniques, Lecture 16
This is a continuation lecture from lecture 15, expanding on different frameworks for pipe-and-filter and different frameworks for the C2 cycle. (Source: John Wiley & Sons)

Applied Architectures, Lecture 17
This lecture illustrates how principles have been used to solve challenging problems. (Source: John Wiley & Sons)

Applied Architectures, Lecture 18… Part 2
This lecture discusses, decentralised architectures, grid protocol architecture and peer to peer architectures. (Source: John Wiley & Sons)

Intro to Domain – Specific Software Engineering, Lecture 23
This lecture discusses concepts domain-specific software engineering (DSSE). The key factors of DSSE domain, business, and technology. (Source: John Wiley & Sons)

Intro to Domain – Specific Software Engineering, Lecture 24
This lecture continues from lecture 23, covering Product lines and relationships between DSSAs, product lines, and Architectural Styles. (Source: John Wiley & Sons)

Standards, Software Lecture 26
This lecture aims to discuss, what standards are and why to use and not use them and prevalent architecture standards. (Source: John Wiley & Sons)

CMMI Documents
CMMI-focused documents have been added from numerous sources, including Carnegie Mellon University (CMMI's originator) and a variety of organizations that consult on this topic or have reported on their journey toward improved capabilities:

Atern and CMMI

Objectives and approach to Atern, CMMI objectives and approach, Comparison of approach, Fitting Atern and CMMI together, Scope and coverage. (Source: LAMRI Limited)

Advanced Process Definition for absolute beginners
This presentation covers basic process definition, advanced process definition, the problem and when to stop. (Source: LAMRI Limited)

CMMI: Introduction
This presentation covers the origins of CMMI, what it is, the premise behind CMMI and its development, model structure and viewing the model levels. (Source: Process Maturity Profile - CMMI SCAMPI Class A Appraisal Results 2007 end-year update, Software Engineering Institute)

CMMI for Recession
This is presentation presents how to maximise the benefits of your process improvement initiative in difficult economic conditions and introduces recession means for your business and the traditional approach to process improvement. (Source: LAMRI Limited)

CMMI for Services (CMMI-SVC)
This presentation covers why CMMI-SVC is needed and how they differ from each other, characteristics of service providers, stakeholders, purpose, and history. (Source: Carnegie Mellon University)

CMMI Level 3 - What the book doesn't tell you!
This CMMI presentation explains what QinetiQ does, SDS operating model diagram, why SDS considered CMMI, and implementation process show though graphs and diagrams. (Source: QinetiQ)

CMMI made practical – Conference London, 2 May 2018
This conference reviews personal and corporate banking IT and operations with approach to implement CMMI. (Source: Carnegie Mellon University)

CMMI OR AGILE?
This conference presentation discusses problem definition, origin of two extremes, factors that affect perception, the truth about CMMI and Agile, seeing the value in both paradigms, problems not solved by CMMI nor Agile. (Source: Entinex, Inc. & DJA & Associates)

CMMI Practitioners: How Can We Improve the Skillset?
This CMMI conference presentation presents CMMI practitioners, CMMI levels 2 & 3 for practitioners and SEI certification. (Source: Carnegie Mellon University)

CMMI version 1.3 Demystified
This CMMI made practical presentation discusses the Aims and Concepts, structural changes, agile development & engineering best practices, other key changes to SGSs and SPs and SCAMPI Version

1.3 changes. (Source: LAMRI Limited)

Concerns with SCAMPI-induced overhead in small settings
This presentation describes the experience in 37 SCAMPI classes A, B and C appraisals, model components, methods, and interface issues. (Source: Carnegie Mellon University)

First field experiences of CMMI-ACQ
This presentation introduces managed acquisition into multiple customer sites, client needs, the proposal, logistics and approaches taken. (Source: LAMRI Limited)

Getting started with CMMI
This presentation covers basics with getting started with CMMI including what it is, benefits of using CMMI and experiences and lessons learned. (Source: LAMRI Limited)

Getting to CMMI L2
This presentation introduces the DFTS (Defence fixed telecommunications service) journey from process methods to communication and more. (Source: BT network)

Implementing CMMI for High-Performance
This CMMI conference presentation covers topics on maturity and performance, software engineering institute support and finding a high-performance improvement solution. (Source: Carnegie Mellon University)

Industrialization Software Engineering using CMMI
This presentation expands on CMMI process framework, maintaining and improving the solution, structure the global process and identifying the delivery process. (Source: Atos Origin, global system integration inc.)

Making the pieces fit
This presentation focuses on making the pieces fit, aligning process and tools, followed by Common Standard Requirements. (Source: LAMRI Limited)

Measuring Capability – A Rational Perspective
This IBM presentation discusses defining challenges in effective software delivery and defining capability improvement framework. (Source: IBM Corporation)

Methodology, tools and CMMI
This requirement management presentations defines a methodology and building a requirement management methodology. (Source: Carnegie Mellon University)

Our CMMI Journey
This presentation describes the CMMI journey undertaken on the BT NHS Spine programme to achieving 18 process areas at capability level 3 and reflection on the route taken, problems encountered, and lessons learnt. (Source: BT network)

Planning Transformational Change
This CMMI made practical presentation presents planning transformational change, covering purpose and objective, workshop unique selling points (USP), key challenges and hosts experiences. (Source: LAMRI Limited)

Practically Level 5
This CMMI made practical conference presentation discusses CMMI high maturity controversy, establishing quality and process performance objectives, process performance models, composingthe defined process and organisational innovation and deployment. (Source: Carnegie Mellon University)

Process and Product Quality Assurance (PPQA) for NASA Ames Research Center
This CMMI presentation discusses the NASA organisation, SS&MA division, software Assurance, CMMI vs. six sigma, NASA Intelligent Systems Division and SCAMPI. (Source: NASA)

Process Optimisation Using the CMMI and the benefits Realised operating at High Maturity
This conference presentation reveals many topics such as the CMMI in Process Development, process Evolution, understanding business processes, systems integration and process inputs and outputs. (Source: CMMI: A Path to Improved Systems, a presentation by Joe Jarzombek, Deputy Director for Software Intensive systems, Acquisition resources and Analysis Directorate, Office of the Under Secretary of Defence (AT&L) on March 2002)

Proof that Process Improvement Works!
This presentation discusses scenarios in attaining high maturity, history of CMMI, graphs relaying organisational processes. (Source: Organizational Systemic Causal Analysis & Resolution Group Metrics Analysis Group)

State of the UK CMMI Market
This paper presents SCAMPI Class A Appraisal Results by Maturity Level and several SCAMPI appraisals through visual diagrams. (Source: Process Maturity Profile - CMMI SCAMPI Class A Appraisal Results 2007 end-year update, Software Engineering Institute)

Thales: DLJ Transformation Programme
This CMMI made practical presentation discusses the overview of Thales, how to change, process improvement, Lean and CMMI working together, implementation strategy, capability and creating a successful environment. (Source: Thales Inc.)

The Art of The Possible
This CMMI made practical presentation introduces improvements needed, investment case control and management of ROI performance, Solution plans to success and results. (Source: LAMRI Limited)

Too small for CMMI?
This CMMI made practical educational presentation outlines how it's possible for a small organisation can use CMMI by introducing the background, the challenges and quick wins, benefits and lessons learned. (Source: LAMRI Limited)

Using CMMI for Services
This presentation covers management, strategy, vision, and stakeholder involvements, evolution model. (Source: LAMRI Limited)

Using TSP to implement CMMI
Topics discussed in this presentation cover implementing CMMI for higher maturity and higher performance, team software processes and implementing CMMI with TSP. (Source: Carnegie Mellon University)

You must be the change you want to see
This presentation discusses what the meaning of CMMI implementation, what should be prepared for, dealing with human change management in your CMMI implementation project, importance of change management and ADKAR model processes. (Source: Primavera business software solutions inc.)

Additional Resources
Additional resources that cover a wide range of systems engineering and related topics include:

Army Acquisition Procedures

This pamphlet provides discretionary guidance on materiel acquisition management. It contains information relevant to research, development, and acquisition, and Life Cycle Management of Army materiel to satisfy approved requirements. This pamphlet major and non-major systems, highly sensitive classified acquisition programs, automated information systems, and clothing and individual equipment. (Source: Department of the Army Pamphlet 70–3)

Change Management & Knowledge
This presentation covers topics on the importance of change management knowledge in the organisation, how change management knowledge can advantage an organisation, the scope and nature of change management in your organisation and how to get effective change in your management. (Source: Thales UK inc.)

Common Anti-Patterns
This presentation looks at the meaning of Anti-Patterns and where they apply and overall process improvement. (Source: Software Engineering Institute)

Dogma, Ego, and Fear - The three Watchmen of Change
Presents unclassified information on process of model systems, waves of human socio-economic development, organizational levels & culture, tools of change and implementation. (Source: Government Communications Headquarters (GCHQ))

Description Document - Annex to draft Statement of Work
To specify the technical requirements and Defence operational concepts for the Supplies to be provided under the Contract. (Source: ASDEFCON (Complex Materiel) Volume 2)

Future challenges for Systems and Software cost estimation and measurement
This conference presentation summarises current and future trends that create challenges for DoD systems and software data and collection analysis, updated software data definitions and estimation methods to help DoD systems and management. (Source: University of Southern California, USC-CSSE)

System behaviours: state machine diagrams
This presentation captures the admissible behaviour system components. (Source: John Wiley & Sons)

Theoretical Foundations for joint Systems Research
This paper proposes a theory of Joint Systems which is intended to be used to assist the Australian Defence Force to enhance its joint war-fighting capabilities. It should be noted that a theory that is suitable for treating Joint Systems, which are composed of human organisations supported by technology, may differ significantly from a typical physical theory in which the human element is absent. (Source: Systems Engineering and Evaluation Centre of the University of South Australia)

The SEG is a free resource, intended for use by clients, alumni and friends of Project Performance International (PPI) as well as clients, alumni and friends of subsidiary company Certification Training International (CTI). If you do not already have access to the Systems Engineering Goldmine, you may apply for free access here.

Updates to SE Tools Database (SETDB)

The Systems Engineering Tools Database (SETDB), developed by PPI in partnership with INCOSE, continues to expand.

Recent SETDB updates, including both new tools and updates to existing tools, include:

Vendor: dSPACE Inc

- Automotive Simulation Models (ASM): Used to simulate combustion engines, vehicle dynamics, electric components, and the traffic environment. The open Simulink models are used for model-based function development and in ECU tests on a hardware-in-the-loop (HIL) simulator.
- SYNECT: A data management and collaboration tool with a focus on the efficient and automated verification and validation of ECU (electronic control unit) software.
- SIMPHERA: A software solution for simulating and validating applications for autonomous driving. The software brings them to life at an early stage, and helps you analyze them and handle their complexity by giving you easy access to powerful functions for simulating.
- SCALEXIO: The modular dSPACE SCALEXIO system can be used for hardware-in-the-loop (HIL) and rapid control prototyping (RCP) applications. It is highly scalable, provides high-performance processor technology for demanding real-time requirements.
- Virtual Validation (VEOS): A PC-based simulation platform for validating the software of electronic control units (ECUs) in early development process stages. A wide range of models - function models, virtual ECUs, bus systems, and plant models - can be simulated with VEOS.

Vendor: DSI international inc.

- Integrated System Diagnostic Design: The Integrated System Diagnostic Design (ISDD) Tool Suite provides all the diagnostic needs of system design. The design process is centered around functional modeling of systems to develop the diagnostics and failure analysis necessary to fully support effective integrated systems design.
- eXpress: A model-based diagnostics engineering software application providing an environment for the design, capture, integration, evaluation and optimization of complex or large-scale system diagnostics, prognostics health management (PHM), systems testability, FMEA and system safety analysis.

Vendor: IncQuery Labs

- IncQuery AUTOSAR UML Bridge: The ultimate solution for Assisted Documentation Creation and Automated Handover. It can streamline your automotive engineering workflow by generating high-quality UML models from AUTOSAR projects, with built-in ISO26262 and ASPICE compliance.

Vendor: OMNINET Software Solutions

- OMNITRACKER Dispatch Center: A dispatch solution intended for enterprises to manage to dispatch field staff flexibly and plan all assignments based on need.

Vendor: Serlio Software

- CaseComplete: A requirements management tool intended for business analysts to capture software requirements and use cases. Once data is captured you can have the system automatically generate requirements specifications, test plans, and project plans.

Vendor: SPEC Innovations

- Innoslate: The first web-based MBSE tool, was developed by SPEC Innovations to support the entire system or product lifecycle. This cloud or on-premise application simplifies

system or product development while reducing time-to-market, cost, and risk.

Vendor: The REUSE Company

- RQA - QUALITY Studio: Copes with the plethora of different engineering items, methods, processes and tools to provides tailored analysis and configurable assessments represented by a centralized system quality scoreboard that provide a quick understanding of the current quality status of a project.
- V&V Studio: Extends quality analysis to all the engineering items generated during a systems engineering life cycle. The requirements quality is managed within logical models (UML or SysML), physical models (Modelica, Simulink) and textual documents.
- RAT - AUTHORING Tool: The perfect assistant when writing requirements, system documentation, or creating models. RAT leads engineers with a set of agreed-upon patterns, ensures the right grammar, and provides real-time quality checking of the elements that are being created.
- SES KM - KNOWLEDGE Manager: Allows you to manage knowledge from the systems engineering point of view and to store valuable information from requirements, models, system architectures and other documents in a common System Knowledge Base.
- SES Requirements Engineering for MS Word: This connector to Microsoft Word allows you to define, measure, improve, and manage the quality of your requirements specifications. It allows assessment of Correctness, Consistency and Completeness (CCC), as well as full traceability of requirements.

Vendor: Tom Sawyer Software

- Tom Sawyer Model Based Engineering: Using Tom Sawyer Model-Based Engineering software, systems engineers can instantly create high-quality, customizable diagrams with automated layout for SysML models. This is essential for in application fields that are being shaped by digital transformation and for progressively complex projects.
- Tom Sawyer Business Process: An advanced business process design and execution application created with your entire organization in mind - from managers, to process modelers and task owners. You can create BPMN 2.0 compliant processes and execute the processes to evaluate precision and consistency of you business processes.
- Tom Sawyer Graph Database Browser: A powerful, easy-to-use web application that instantly visualizes the data in your graph database. It's designed for data scientists, analysts, architects, and developers to see and analyze graph databases in any browser, without programming.

Vendor: Verocel Inc.

- VeroTrace: VeroTrace is a cloud-based application lifecycle management tool that provides control and management of software lifecycle data from requirements, design, code, test cases and results as well as all associated documentation. It also supports compliance with many certification standards.
- VeroSource: VeroSource automates the capture and structural coverage analysis of the source code under test.
- VeroSource-A: verifies that all complex decisions (i.e., decisions with at least two conditions) have taken all possible outcomes; and that every condition within the

decision has taken all possible outcomes that have independently affected the decision's outcome.
- VerOCode: A unique capability that provides a coverage analysis and mapping between the object code executed on the target and the associated source language statements. It directly satisfies the ISO26262, SIL4 and DO-178B/DO-330 TQL-5, Level A MC/DC object coverage objectives for code coverage.

Vendor: Vitech

- CORE: Built upon a proven systems metamodel, CORE delivers integrated support for model-based systems engineering from first concept through definition of behavior, physical architecture, and V&V with all project information residing in a coherent repository delivering an effective system source of truth.
- GENESYS: Built upon a proven systems metamodel, GENESYS delivers integrated support for model-based systems engineering from first concept through definition of behavior, physical architecture, and V&V with project information residing in a coherent repository delivering an effective system source of truth.

SyEN readers are encouraged to check out these new and updated systems engineering tool offerings.

Access the SETDB website.

CONFERENCES, MEETINGS & WEBINARS

ProjectWorld - BusinessAnalystWorld Conference

The International Institute of Business Analysis™ (IIBA®) is a professional association with over 30,000 members that helps the global business analysis community to achieve better outcomes through better analysis. IIBA® has endorsed the ProjectWorld – BusinessAnalystWorld conference (PW-BAW 2023) conference that will take place in Toronto, Ontario, Canada from 15-18 May 2023. The theme of this in-person conference is "*Future Forward*". The PW-BAW 2023 conference is focused on educating project managers and business analysts in concepts and skills that are critical to their disciplines and jointly beneficial.

Beyond the keynotes, the education focus of the conference is evident in the mix of 30+ one-hour presentations and 20 full-day workshops that enable attendees to gain 24 hours of key learnings (and the associated PDUs/CDUs).

Presentation topics include:

- *The seven pitfalls of business analysis.*
- *7 mindset shifts for success in digital transformation.*
- *The neuroscience behind the BA mindset. The human thinking processes applied.*
- *Cross-team collaboration is hard! Tips & tricks to help solve divides.*
- *Bamboozled! Are your requirements psyched out? Using critical thinking for better decisions and requirements.*
- *How can project managers use data to make smarter project decisions!*

Workshops will address such topics as:

- *Be a digital savvy leader.*
- *Mitigating risks in everyday projects.*
- *The who, what, when, where (and how!) of requirements.*
- *Building better product owners.*
- *Data management & analytics skills for business analysts.*
- *Leveraging Design Thinking to drive value.*
- *Long term Agile planning is never pointless.*

Download the conference brochure.

Learn more and register.
Learn more about IIBA.

CONFERENCES, MEETINGS & WEBINARS

NAFEMS World Congress 2023 (NWC23) Agenda

The detailed agenda has been released for the NAFEMS World Congress 2023 (NWC23) to be held on 15-18 May 2023 in Tampa, Florida, USA. The conference theme is *A World of Engineering Simulation.*

NWC23 will feature 300+ presentations arranged in multiple concurrent tracks. The conference will deliver 15+ workshops and 11+ short training courses.

See the detailed agenda here.

The keynote speakers and their topics have been updated and now includes:

- Mechanics Meets Biology: Modeling and Simulation Towards Skeletal Tissue Regeneration (Sara Checa, Julius Wolff Institute Berlin Institute of Health)
- The History of the Iconic Boeing 747 and the Evolution of Simulation Utilization Over its Development (Steven Chisholm, Boeing Commercial Airplanes)
- The Science and Mission of the James Webb Space Telescope (Michael Menzel, NASA)
- A Half-Century of Research and Mentoring in Fluid Dynamics from Hemodynamics to Hypersonics (Wesley L. Harris, MIT)
- Digital Transformation of System Performance Development by a Flexible Digital Thread (Ernesto Mottola, Toyota Motor Europe)
- Developments in Advanced, Physics-based Modelling and Simulation Techniques in the Realm of Defence and Security at Dstl (Daniel Pope, Defence Science and Technology Laboratory (Dstl))
- The Role of Modeling and Simulation in the Age of AI (Mahmood Tabaddor, Accenture)

Register for NWC23.

Learn more about NAFEMS.

PTC LiveWorx 23 Conference

PTC, a leading Product Lifecycle Management solution provider, is hosting the PTC LiveWorx 23 Conference in Boston, Massachusetts, USA from 15-18 May 2023. The theme of this in-person conference is "*A New Era in Product Lifecycle Innovation*". The conference will address innovations in both product lifecycle methodologies and enabling technologies in over 200 sessions in 7 tracks:

- Application Lifecycle Management (ALM)
- Augmented Reality (AR)
- Computer Aided Design (CAD)
- Environmental Sustainability
- Internet of Things (IOT)
- Product Lifecycle Management (PLM)
- Service

Learn more about LiveWorx 23 keynote presenters and track spotlight speakers here.

View and search the detailed agenda.

The XTROPOLIS technology showcase at LiveWorx 23 will provide attendees with opportunities to view and interact with a wide range of Digital Transformation technologies for engineering, manufacturing, and service.

LiveWorx 23 will also include a Service Lifecycle Management Symposium that runs in parallel with the main conference.

Learn more about LiveWorx 23.

Register here. Note that there is a free Digital Pass registration option that enables access to livestreams of the conference keynotes and track spotlights.

View selected content from LiveWorx 21.

Front End of Innovation (FEI 2023) Conference

From 16-18 May 2023, Boston, Massachusetts, USA will be the site of the 20th Front End of Innovation (FEI 2023) conference. The theme of FEI 2023 is "*Explore & Exploit*". Participants will have the opportunity to learn from innovation and thought leaders in numerous fields such as consumer packaged goods, healthcare, pharma, financial, academics, and creatives. Over 80 presentation sessions will be organized into seven tracks:

- PEOPLE: What does it take to be a great innovator/innovation team? This track will focus on us as innovators, the mindset, team building, culture, future goals and career!
- FORESIGHT: What do our customers want? This track will focus on trends and foresight practices for meaningful innovation.
- STRATEGY: How do we strategize for big breakthroughs? This track will focus on frameworks for creating high revenue innovation across large organizations.
- DESIGN: Trends in design as well as design led innovation as means to developing organizational capabilities to respond to changing markets.
- DATA: What does data tell us about the future of innovation?
- GROWTH: What are the commercially minded innovations of the future? In this track speakers will focus on the ROI and innovation acceleration.
- PARTNERSHIPS & ECOSYSTEMS: What are the secrets of a perfectly orchestrated startup and big corporation ecosystem? This track will focus on successful and impactful cooperation between the two.

Topics to be addressed include:

- Innovation Mindset Adoption across an Organization
- Accelerating Revenue Growth in Innovation
- Leading Innovation with Design
- Foresight Practices for Meaningful Innovation
- What is the Metaverse, and Why Should I Care?
- Ambidextrous Leadership
- Organizing for Disruption
- How to Pick your Next Big Idea

Learn more about FEI 2023.

View the detailed agenda.

Download the conference brochure.

Register for FEI 2023.

IISE Annual Conference & Expo 2023 Program Details

The Institute of Industrial and Systems Engineers (IISE) has released the detailed program for the IISE Annual Conference and Expo 2023 to be held in-person in New Orleans, Louisiana, USA on 20-23 May. PPI is proud to be exhibiting in person at the IISE event! If you will be attending in person too, PPI's Randy Iliff and René King would be happy to say hello! View the program details here. A small sample of interesting topics and associated presentations (with abstracts) for SE practitioners includes:

Systems Engineering in Manufacturing:

- Systems Engineering Application on Build to Print Programs: The Systems Engineering (SE) discipline is typically correlated with the Design cycle on a program. However, the power of the SE toolbox goes well beyond design requirements and verification. This presentation will run through an example of a Build to Print program and how SE tools helped keep focus on Customer Requirements through program set up. This was achieved by carefully processing all of the contractual documents and extracting requirements. By doing so, the setup team was able to identify gaps from customer requirements to current Spirit processes, and help build plans to close those gaps. (Joan Wager – Spirit AeroSystems)
- Systems Engineering Approach to Production Design: The Systems Engineering discipline has long been applied to the design of product. With successes being numerous and well documented for reduction of cost through the product life cycle by following these processes, the next logical step is applying the skillset to design of production processes. This presentation will discuss several applications of how incorporation of SE into the early phases of production planning as well as acting as a mediator with product design, can minimize cost and disruption through the startup phase of production. (Joan Wager – Spirit AeroSystems)
- Manufacturing Readiness Level Tracking Within the Defense Industry: A manufacturing readiness level (MRL) is a metric used within the defense industry to measure a program's production capability. The current process used to track MRLs is manual and requires an enormous amount of manpower and resources to produce a final report that can be delivered to the customer. A program's MRL is determined by multiple parts of the manufacturing process and their coordinated measurements such as yield rate, labor hours, supplier delivery targets, etc. Each of these measurements must be individually obtained from various parts of a company, which leads to a lengthy process. The scope of this research project is to directly address process changes needed to reduce cycle time regarding the MRL process. Currently, production is tracked within a system that supports manufacturing operations, quality tests, material procurement, and more. Through this research project, it was found that this existing system could be expanded to support an automated MRL rating tool. The integration of this tool provides easily accessible information for upper management and supports improved customer responsiveness. The need to quickly assess manufacturing readiness is a common need in industry and this research provides a case study that shows how improving the process of performing a correct MRL assessment improves overall operations. (Hannah Lavier and Morganne Murphy, California Polytechnic State University)

Systems Engineering and Modeling

- A Small-Scale Simulation Model of an Emergency Department: Rising patient volumes in emergency departments (EDs) around the world are putting significant strain on both Emergency Medicine and entire healthcare systems. Emergency departments (EDs) are many times the first points of entry into hospitals for patients in both normal and severe health situations. One of the ED's primary challenges is to meet the unpredictable patient demand. Several factors influence patient wait time in the emergency department. To better understand the factors influencing the ED's ability to meet patient demand, we propose a Discrete Event Simulation (DES) model. Based on data collected from job shadowing at Phelps Health Emergency Department in Rolla, Missouri, a small-scale simulation model was constructed. The simulation model represents specific aspects of the ED system's structure, resources, and actors. The simulation model considers various ED factors such as rate of arrival, acuity level, and resource availability. The effect of these factors on the average length of patient visits at various times was investigated and recommendations for ED managers were developed. (Prachita Humane, Missouri University of Science and Technology)
- A Multi-Criteria Ranking System for Prioritizing Maintenance of Levee Systems in Arkansas: There are 208,009 properties in Arkansas that have more than a 26% chance of being severely affected by flooding over the next 30 years, which represents 13% of all properties in the state. A levee system is generally designed to reduce the flooding risk for urban and rural communities; however, most of the state's levees have been significantly outdated or built with engineering standards less rigorous than current best practices. The Levee Safety Action Classification (LSAC), as recorded in the National Levee Database (NLD), communicates the risk associated with living behind a particular levee and assists local, state, and federal stakeholders in identifying and prioritizing funding needs. It is expected that LSAC rating will decrease as flood risk decreases. However, in some cases LSAC rating for a particular levee may remain the same despite its performance or infrastructure health when the leveed area is densely populated or significantly developed. We develop a multi-criteria ranking framework, integrating Principal Component Analysis (PCA) and Multi-Criteria Decision-Making (MCDM) methods (i.e., a CRITIC-TOPSIS approach), for prioritizing maintenance of the levee systems in Arkansas using the NLD data. Additionally, we perform a cost-benefit analysis, comparing the operating and maintaining costs with the associated benefits to determine maintenance prioritization. (Nguyen Phan, University of Arkansas)
- Virtual Reality Applications and Scenarios in Project Management and Medical Field: Technology has revolutionized the field of project management and medical care. In the past, project management was a cumbersome and time-consuming process that relied heavily on manual labor. Project management software and electronic medical records (EMRs) make the process much more streamlined. The adoption of V.R. as a supplement to medical training would greatly benefit patient care. Virtual reality (V.R.) can train doctors to perform laparoscopic procedures with greater precision. The use of virtual reality technology "is quite helpful in transferring abilities to the operation room." The American Board of Internal Medicine recommends that medical residents use virtual reality (V.R.) equipment to practice procedures on virtual patients before attempting them on actual patients. The potential applications for virtual reality technology are vast and varied. Intertwining its complexities in the project management and medical field increases the possibility of envisaging a host of novel and exciting ways to use V.R. While its potential remains largely untapped, there are several compelling reasons why V.R. should be integrated into project management and medical practice. Health institutions must open up

new and innovative ways to care for their patients, while project managers can utilize V.R. to manage complex projects more effectively. (Udit Sai Anand Nalukala and Sohyung Cho, Southern Illinois University Edwardsville)

- Case Study Analysis of Sustainable Development using Cluster Analysis and Systems Dynamics: Current sustainability models lack frameworks that integrate theoretical reasoning into implementable use cases. This paper uses case study analysis focused on sustainable development. Three distinct contributions are made in this work as evidenced by the examined cases. The first contribution identifies mediating factors that should be included in any framework to describe sustainable development. The second contribution uses neural network applications to validate the model by ensuring the indicator data can predict the target sustainable development score. The third contribution creates a new framework for sustainable development models that considers other factors that impact sustainability outside of social, environmental and economic influences. Engineering managers can use these results to implement strategies to achieve specific sustainability development goals. (Tiffanie Toles, Missouri S&T)

Other topics of interest might include:

- Human Factors in Human Performance
- Sustainable Development Goals (SDGs)
- Facility Design
- Student Success
- Intelligent Transportation Systems
- Simulation Models in Healthcare Management
- Systems Engineering in IE Practice
- Industry 4.0 and Smart Manufacturing – Challenges and Opportunities
- OR Innovations in Disaster Management
- Advanced Tissue Engineering: Innovations and Applications
- Flexible Design Case Studies

Learn more. Register here.
Join IISE.

International Conference on Axiomatic Design (ICAD2023)

Axiomatic Design

Axiomatic Design (AD) is a rule-based product design methodology that was developed by Professor Nam P. Suh at Massachusetts Institute of Technology (MIT) in the 1970s. AD is a systems engineering (engineering design) methodology that uses matrix methods to systematically analyze the translation of customer needs into functional requirements, design parameters, and process variables. The two axioms of Axiomatic Design are: maximize the independence of the functional requirements and minimize the information, or complexity of the system/solution.

The 15th International Conference on Axiomatic Design will be held on 31 May – 2 June 2023 in Eindhoven, The Netherlands. The conference will focus on the application of AD to Industry 4.0, the design of intelligent systems and design education.

Presentation topics include:

- *Advantages in Axiomatic Design Theory*

- *Application and Case Study of AD*
- *Complexity and design of Complex Systems*
- *AD for Software Design and Development*
- *Integration of AD with other Design Methods*
- *AD in Social Science & Management*
- *AD for Engineering Education*
- *AD for Manufacturing*
- *AD for Sustainability*
- *AD for Tolerance & Design*
- *The Future of AD*

Learn more about ICAD2023. Check here for updates to the conference program.

Access publications from previous ICAD events:

- ICAD Papers 2000-2016 (includes recommended books on AD theory and practice)
- ICAD Papers 2017 (Science Direct)
- ICAD Papers 2018
- ICAD Papers 2019

Integrate23 Agenda and Registration

The detailed agenda and registration are available for the Integrate23 conference to be hosted by Zuken Vitech in Scottsdale, Arizona, USA, on 5-8 June 2023. The conference will offer diverse presentations on topics in digital engineering, systems engineering and enterprise architecture.

Integrate23 will kick off on 5 June with System Architecture Modeling Competition in which participants will exhibit their skills and creativity in designing and constructing a systems solution for a real-world problem. Participants have the liberty to select their preferred modeling language, methodology and tool.

The detailed agenda indicates that content has been arranged into four role-focused tracks with presentations shown below for each track:

For the Designers

- Systems Engineering Challenges of the Next Generation Space Suit
- Digital Standards with MBSE
- Six Fundamentals of a Successful MBSE Implementation
- Understanding the Digital Thread between MBSE and Programs Risk Management
- SysML2 in GENESYS
- Model Based Mission Assurance
- The UAF in GENESYS
- Better Living Through Architecture: How Investing in Architects Benefits the Team
- Automated Quality Assurance for Next-Generation Digital Threads
- Incorporating Artificial Intelligence in the Systems Engineering Process
- Risk Management as an Integral Element of Model-Based Systems Engineering (MBSE) using Case Studies and Examples

For the Makers

- Engineering the Digital Ecosystem
- New Digital Thread Solutions for Architecture and ECAD Models
- Quantitative FMA: Using Coupled MBSE and Simulation Methods to Remove the Guesswork from Failure Mode Analysis
- MBSE 360
- Developing Parallel Programs While Maturing and Implementing Digital Engineering Capabilities
- Safely Navigating the River of Doubt: Executing a Successful Digital Transformation
- Digital Engineering: An Engineering Problem with an IT Solution
- Complexities of Open Digital Thread for Commercial and Government Aerospace Programs
- Impact Analysis of System Architecture for Automotive Systems
- MBSE Stack DevSecOps
- Thoughts on Teaching MBSE
- Open MBEE Update

For the Thinkers

- Ontology-first hub and spoke Architecture for Model Interoperability
- A Model for Cyber Mission Assurance in an MBSE Framework
- Empowering Digital Collaboration Through the Adaptive Enterprise Framework
- Digital Engineering Application of the Environment-Vulnerability-Decision-Technology Framework for Hurricane Disaster Relief Decision Support
- Model Based Systems Engineering Rapid Response Process
- Maturating decision-making on Industrial Assets by using Digital Assets
- How to Assess Your Current MBSE Maturity and Develop a Roadmap to Meet Your Targets
- Design, Verification and Validation of an Avionics Cable Assembly using SysML
- The GitHub revolution is coming to MBSE. Are you ready?
- What's New in GENESYS
- An Approach to Digital Engineering using the Common Information Infrastructure (CII)

For the Innovators

- Assessing Methods to Increase the Value-Added Interplay Between MBSE and Detailed Design Tools
- Design for Sustainability Developing a Design Optimization Culture
- Introduction to the Integrate23 Digital Engineering Hub
- Accelerate Development and Increase Productivity Through Digital Transformation
- Configure, Select, Tune. Trade Space Analysis & Simulation for a System Family
- How Lockheed Martin Streamlined Their Digital Thread With DS2
- Roadmap for PCB Design Using Artificial Intelligence
- Practical Applications of Implementing a Digital Thread
- SBE Vision and Genesys on the Digital Thread
- Model-Based Systems Engineering is an Evolving Necessity, Delivering Value Into the Product Design Process
- The PCB Subsystem: Architecture Requirements Analysis & Traceability Gate
- Visualizing Sandia's System Engineering Architecture Using GENESYS and Python
- Development of a System Integration Tool Leveraging a Commercial Game Engine

Register for Integrate23.

CONFERENCES, MEETINGS & WEBINARS

Registration Open for International SoSE Conference

IEEE System, Man, and Cybernetics Society (SMC) will host the 18th International Conference on System of Systems Engineering (SoSE) in Lille, France on 14-16 June 2023. The theme for this hybrid conference is *AI and Autonomous Robotics in System of Systems. INCOSE is a co-sponsor for this conference.*

Register for SoSE 2023 through the EDAS system.

Look for program updates and learn more about SoSE 2023.

Program Updates for 10th Resilience Engineering Symposium

RESILIENCE ENGINEERING 10

26-30 JUNE 2023, SOPHIA ANTIPOLIS FRANCE

A detailed program has been released for the 10th Resilience Engineering Symposium, to be held in Sophia Antipolis, France on 26-30 June 2023 to support the theme of "*Resilience at frontiers, frontiers of resilience*".

Resilience Engineering is a trans-disciplinary perspective that focuses on developing on theories and practices that enable the continuity of operations and societal activities to deliver essential services in the face of ever-growing dynamics and uncertainty. It addresses complexity, non-linearity, inter-dependencies, emergence, formal and informal social structures, threats, and opportunities.

A sample of the topics to be addressed in the full program includes:

- Serious Games for Resilient Infrastructure Systems
- How to Build and Sustain Adaptive Capacity in Turbulent Worlds: Putting the Theory of Graceful Extensibility to Work
- Coordination at scale under time pressure, uncertainty, ambiguity and innovation
- Industry contributions - Resilience professional sharing experience
- New perspective – Resilience visualization
- Digital technologies and AI for resilience and adaptive capacity
- Surprises and resilient societies
- Designing for resilient performance
- Limits of Resilience Engineering

View the detailed agenda here.

Call for Papers & Tutorials: TdSE 2023

TdSE Tag des Systems Engineering

INCOSE's German chapter, GfSE (Gesellschaft für Systems Engineering e.V) is hosting its annual conference, Day of Systems Engineering (TdSE®), in Würzburg, Bavaria, Germany from 15-17 November 2023 with its motto as "*Future needs courage!*"

The Call for Papers and Call for Tutorials have been released, seeking contributions in the following subject areas:

- Systems engineering for a sustainable future.
- Systems Engineering and Artificial Intelligence.
- Systems engineering in infrastructure projects (e.g., municipal administration, rescue

services, public institutions).
- Systems engineering in infrastructure projects (e.g., municipal administration, rescue services, public institutions).
- Systems thinking as a core competency.

Related topics such as systems engineering management, organizational design, modeling, maturity models, and SE in education/training will also be considered.

The deadline for submissions is 19 June.

Learn more about GfSE here.

Free NAFEMS Webinars in May - June

NAFEMS, the international modeling and simulation association, is offering two free webinars in May and June 2023.

Harnessing the Power of Modelling & Simulation in the Food & Drinks Industry (3 May)
The food and drinks industry is worth over $5 trillion globally and involves complex processes and technologies. Engineering modelling and simulation (M&S) can optimize every step of the process from equipment design to product manufacturing, packaging, and distribution, saving time, money, and resources. NAFEMS' newly formed Food and Drinks Industry Community (FDIC) will provide guidance on ways of working towards solving industry-specific problems and introduce the community during their first webinar. The webinar is a prequel to a face-to-face event being held in the Autumn in Birmingham, UK where the community will share experiences, examine and solve unique industry-specific problems, and create a vibrant community of analysts and designers. Register here.

Handling Flexible Bodies in Multibody Dynamics (20 June)
Multibody Dynamics (MBD) simulation is becoming increasingly important in understanding the relationships between actuators, control systems, and final performances of mechatronic systems. While rigid multi-body simulation is suitable in situations where bodies do not undergo significant deformations, flexible multi-body simulation is necessary when compliance induces appreciable deformations. Modal reduction methods have been used for many years to implement flexible bodies in multi-body simulation, but they may provide unreliable results in situations of large deformations or other non-linearities. The best approach is to describe the finite element problem within the multi-body dynamics equations, but only a few commercial tools currently adopt this method. This webinar provides an overview of these topics and best practices for including flexible bodies in multi-body simulation. Register here.

Investigate membership in NAFEMS.

FEATURE ARTICLE

SysML v2: Highlighting the Differences with SysML v1

by

Sanford Friedenthal (SAF Consulting). Email: safriedenthal@gmail.com

and

Ed Seidewitz (Model Driven Solutions), Email: ed-s@modeldriven.com

Version 1.0, April 2023

Abstract

The OMG Systems Modeling Language™ version 1.0 (SysML®) final specification was adopted by the Object Management Group (OMG) in 2007. SysML has been used by many organizations to support their efforts to transition to a model-based systems engineering (MBSE) approach. The next generation Systems Modeling Language, SysML v2, has been submitted to the OMG for adoption as a beta specification. The specification then enters the finalization phase and is expected to become a final adopted specification in 2024. This paper highlights some of the differences between SysML v2 and SysML v1, and the motivation for these differences. This paper is a follow-up to an earlier paper published in this same journal (reference 9) which provided the background and introduction to SysML v2.

Introduction

This paper highlights some of the differences between SysML v2 and SysML v1 and the motivation for these changes.

The OMG Systems Modeling Language (SysML) v1 is a general-purpose graphical modeling language used to model systems that contain hardware, software, data, procedures, people, and facilities. It enables the application of MBSE by providing a standard language to represent many aspects of a system including its requirements, behavior, structure, and parametric constraints. The system model in SysML captures these different aspects of the system and is a primary artifact of an MBSE approach.

SysML version 1.0 was formally adopted by the Object Management Group (OMG) in 2007. The language continued to evolve through a series of incremental versions leading to the release of SysML version 1.7 in December 2022. During this fifteen year-period, much was learned about applying MBSE with SysML which has revealed the strengths and weaknesses of the language.

SysML v2 is the next generation systems modeling language that is intended to provide substantially improved capability for modeling systems over SysML v1. SysML v2 is being developed in response to

the requirements in the SysML v2 request for proposal (RFP) that was issued by the OMG in December 2017, and the SysML v2 API & Services RFP which was issued by the OMG in June 2018. The language specification for SysML v2 was submitted to the Object Management Group on February 20, 2023, for approval as a beta specification. The beta specification enters the finalization phase and provides tool vendors that are implementing the specification an opportunity to submit issues for resolution prior to issuance of the final adopted specification which is anticipated in 2024.

The objectives for SysML v2 are to facilitate improvements in both adoption and effectiveness of MBSE when using SysML v2 versus SysML v1. In particular, the objectives are to improve the language in the following ways:

- Precision of the language leading to less ambiguity
- Expressiveness of the language
- Consistency across the various aspects of the language
- Interoperability with other engineering models and tools
- Usability by both developers and consumers of models
- Extensibility for domain specific applicationsSummary level changes in SysML v2

The following are the summary level changes in SysML v2 relative to SysML v1 that provide the basis for achieving the SysML v2 objectives:

Metamodel

SysML v2 contains a new metamodel that extends the Kernel Modeling Language (KerML), which was developed as the syntactic and semantic foundation for SysML v2. SysML v1 was based on the UML metamodel, which was originally developed for software modeling. The SysML v2 and KerML metamodels were architected to leverage many of the UML modeling capabilities along with additional capabilities to address the needs for modeling systems. The SysML v2 metamodel provides a formal semantic underpinning, consistent terminology, and improved support for modeling deeply nested hierarchies of structure, behavior, and requirements and their cross-cutting relationships. It also provides a library-based mechanism to further extend the language for domain-specific applications.

Graphical and textual syntax

SysML v2 can be expressed using a standard textual syntax in addition to a graphical syntax, whereas SysML v1 can only be expressed using a standard graphical syntax. The complementary ways for expressing the model in SysML v2 can improve understanding and ease of use. In addition, SysML v2 provides a flexible view and viewpoint mechanism and formal specification of the graphical and textual syntax to enable further customization of the views to meet the needs of the stakeholders.

Application Programming Interface (API)

SysML v2 includes a standard API and an associated standard set of services to navigate, query, and update the model. This enables interoperability with other tools and software applications throughout the life cycle of a system development.

General changes in SysML v2 language

The following section describes some of the changes in SysML v2 relative to SysML v1 that apply across the language.

Note: *Examples of the SysML v2 textual syntax in this paper were created using the open-source pilot implementation that was developed as part of the SysML v2 submission development effort. The graphical views of the SysML v2 model were created using a prototype visualization tool integrated with the pilot implementation, based on an open-source application called Plant UML. The quality of the graphical visualization is limited but will be substantially improved when commercial tools become available.*

Textual notation

A major change when modeling with SysML v2 is the introduction of a textual syntax. The textual syntax includes a standard expression language to represent logical and quantitative expressions that enable solvers to unambiguously interpret SysML v2 models. The textual syntax can also be used to exchange models.

The textual and graphical syntax provide different renderings of the same underlying model (e.g., the abstract syntax and semantics). The use of the textual syntax and the graphical syntax provide complementary ways to create, modify, and view the model. The textual syntax is often an effective way to create, view, and modify small, detailed portions of the system model such as a calculation with expressions or a detailed algorithm. At the same time, the graphical syntax is often an effective way to create, view, and modify cross-cutting architectural views of the system including interconnections, allocations, and interactions.

An example of a textual representation for a simple vehicle model is shown in Figure 1. The textual notation shows a *vehicle* that has an attribute *mass*. The vehicle is composed of two parts, an *engine* and a *transmission* which each have an attribute *mass*. The vehicle *mass* is the sum of the *engine.mass* and the *transmission.mass*.

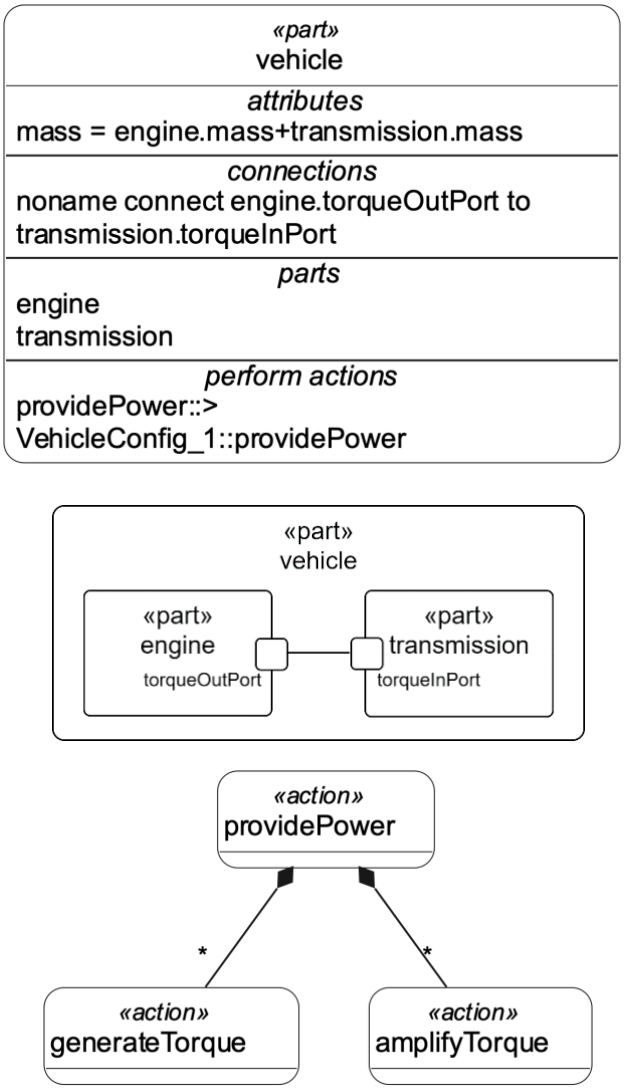
```
part vehicle{
    attribute mass = engine.mass+transmission.mass;
    perform providePower;
    part engine{
        attribute mass;
        port torqueOutPort;
        perform providePower.generateTorque;
    }
    part transmission{
        attribute mass;
        port torqueInPort;
        perform providePower.amplifyTorque;
    }
    connect engine.torqueOutPort to transmission.torqueInPort;
}
action providePower{
    action generateTorque;
    action amplifyTorque;
}
```

Figure 1. Simple vehicle model expressed using SysML v2 textual syntax

The *engine* and *transmission* have ports *torqueOutPort* and *torqueInPort* respectively that are connected as shown by the connect statement.

The *vehicle* also performs an action *providePower* that refers to the *providePower* action below which is composed of the *generateTorque* and *amplifyTorque* actions. The *engine* and *transmission* each perform actions to *generateTorque* and *amplifyTorque* respectively.

The corresponding graphical views for the *vehicle* part, the interconnection between the *engine* and *transmission*, and the decomposition of the *providePower* action are shown in Figure 2. As the model becomes more complex, it is evident that the graphical views are critical for understanding the complex relationships that span the system model, while the textual syntax enables a close-in look at selected portions of the model.



Figures 2. SysML v2 graphical views of simple vehicle model

Definition and usage pattern

Another major change to SysML v2 is the consistent application of the definition and usage pattern. The definition and usage pattern is intended to enable reuse by creating usage elements that are defined by a common definition element. A simple example is a definition of a *Tire* where the front tires and the rear tires are both usages that are defined by *Tire*. However, the usages for front and rear tires can vary in terms of their tire pressure, tire wear, and their connections to different parts of the vehicle. The ability to define a common definition element while adapting each usage to its context can significantly reduce the modeling effort and the potential for introducing modeling errors.

SysML v1 introduced the definition and usage concept informally. For example, a block is considered a definition element (i.e., the type) and a part property is considered a usage element that is typed by a block. Similarly, an action is considered a usage of an activity. However, the actual relationship

between an action and an activity is quite different from the relationship between a part property and a block. In addition, the usage cannot be modified directly to adapt to its context. For the front and rear tire examples referred to previously, the block *Tire* must be specialized to be a *FrontTire* and a *RearTire* each having their own specialized features as shown in Figure 3a. The parts for the front tires and rear tires are then typed by these respective blocks.

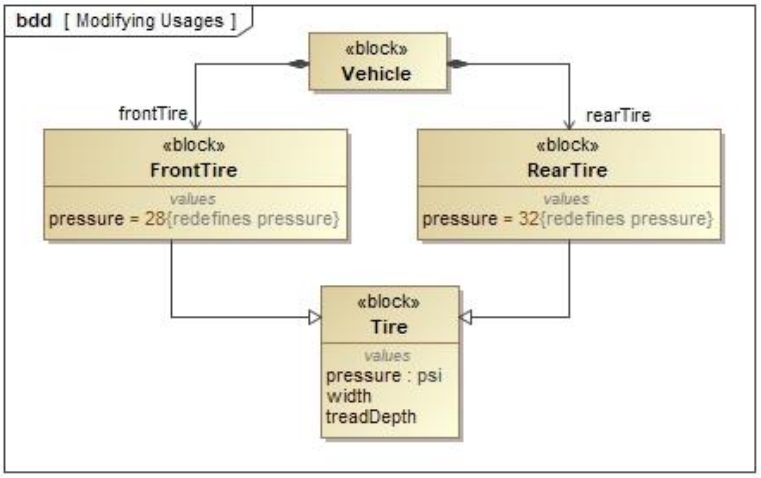


Figure 3a. SysML v1: front and rear tires modified to have different pressures

In SysML v2, the definition and usage pattern is designed into the metamodel at the root level. Virtually all elements in SysML v2 follow a consistent pattern of definition and usage. The relationship between a usage and its definition is a kind of specialization. This enables a usage to inherit the features from its definition which can then be redefined or subsetted, and new features can be added. This aspect of the language makes it more straightforward than SysML v1 to modify a design configuration and/or adapt it to its context. The usages for the front tire and rear tires in Figure 3b have been directly modified to modify the tire pressure. The carrot symbol (^) indicates the *treadDepth* and *width* are inherited from the definition but have not been modified.

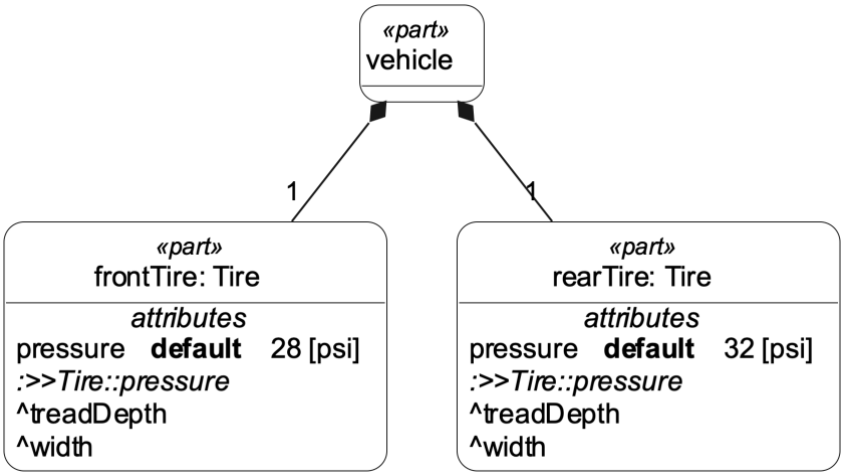


Figure 3b. SysML v2: front and rear tires modified to have different pressures

The terminology is also consistent with this definition and usage pattern. For example, a part usage is defined by a part definition, an action usage is defined by an action definition, and a requirement usage is defined by a requirement definition. The key words are shortened to further simplify their use. For example, the key word for part usage and part definition is **part** and **part def** respectively.

Table 1 highlights some of the SysML v2 concepts that apply the definition and usage pattern, and the corresponding concepts and terminology in SysML v1.

Table 1. Contrasting regular SysML v2 definition & usage pattern with SysML v1 concepts

SysML v2	SysML v1
part / part def	part property / block
attribute / attribute def	value property / value type
port / port def	proxy port / interface block
action / action def	action / activity
state / state def	state / state machine
constraint / constraint def	constraint property / constraint block
requirement / requirement def	requirement
connection / connection def	connector / association block
view / view def	view

The regular application of the definition and usage pattern to virtually all definition and usage elements in SysML v2 including structure, behavior, and requirements and the capability it provides enhances reuse and usability and facilitates machine interpretation of the models.

Decomposition

In SysML v1, a system decomposition is accomplished by defining a block for the system. The block contains part properties corresponding to the components of the system. To further decompose the system, the components must be typed by blocks, and those blocks must contain the next level of part properties. This applies to any level of nesting. The example in Figure 4a shows two levels of nesting for a *Vehicle* that is composed of a *leftWheel* and a *rightWheel* both typed by *Wheel*, and the *Wheel* is composed of *lugbolt1* typed by *Lugbolt*.

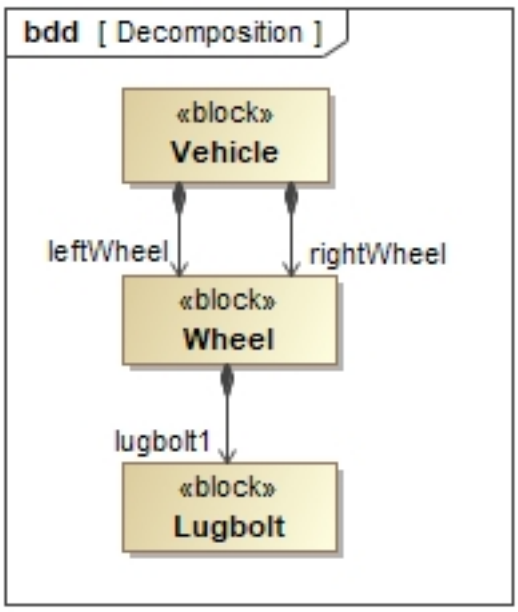


Figure 4a. SysML v1: Vehicle with 2 levels of decomposition

In SysML v2, usages can be decomposed directly so that a system can contain parts that contain other parts which can be further decomposed to any level of nested parts. This is generally a more natural way to represent a system decomposition. Analogous with the SysML v1 model in Figure 4a, the SysML v2 model in Figure 4b represents a vehicle part that is composed of a *leftWheel* and a *rightWheel* and each wheel is composed of a *lugbolt1*.

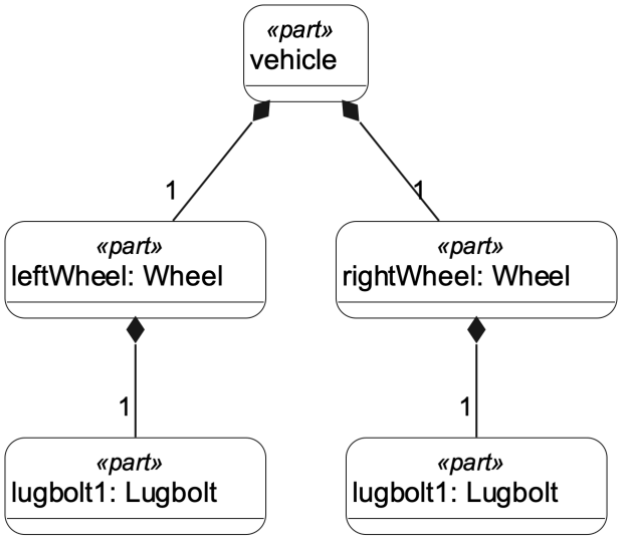


Figure 4b. SysML v2: The vehicle part decomposition is more intuitive and reduces ambiguity

The SysML v2 part decomposition reduces ambiguities that occur in SysML v1 when parts are nested more than one level deep. For the SysML v1 model in Figure 4a, the *lugbolt1* on the *leftWheel* is indistinguishable from the *lugbolt1* on the *rightWheel*. However, there is no ambiguity with the SysML v2 model in Figure 4b where the *leftWheel* and *rightWheel* each have their respective *lugbolt1*.

Membership relationship versus containment

In SysML v1, a package contains packageable elements such as other packages, types, and dependencies. However, this meta-relationship does not appear explicitly in the model browser, but it does show on a diagram as a line with a crosshair on the owner end. In SysML v2, this relationship is a concrete relationship called a membership relationship that is explicitly part of the model. This relationship is not limited to packages but applies more generally to relate any namespace to its members. This results in the SysML v2 model having a true graph-like structure that enables navigation and other benefits. The downside is that it increases the number of relationships the model must store but this is not necessarily a significant issue with today's computational and memory capacity.

In SysML v1, a decomposition relationship is modeled with an association. In SysML v2, a decomposition relationship is modeled as a specialized kind of membership called a feature membership. A part or part definition that is composed of other parts is related to each of its nested parts by a feature membership relationship. The relationships shown in Figure 4b are feature membership relationships and not associations.

The feature membership can be composite or referential analogous to a composite or reference association in SysML v1. A part or part definition can contain a reference part that is bound to a part that is being referred to. For example, consider a trailer that is hitched to a car. The trailer hitch is composed of a *hitchBall* that is mounted on the *vehicle* (i.e., car) and a *trailerCoupler* that is mounted on the *trailer*. The *hitchBall* and the *trailerCoupler* are part of the *trailerHitch* but can be considered reference parts of the *vehicle* and the *trailer* respectively. In this case, the *hitchBall* on the vehicle and

the *trailerCoupler* on the trailer are bound to the *hitchBall* and the *trailerCoupler* on the *trailerHitch* as shown in Figure 5a. The binding connection is annotated with an '=' symbol. This is the same way reference parts are bound to a part in a SysML v1 internal block diagram as shown in Figure 5b.

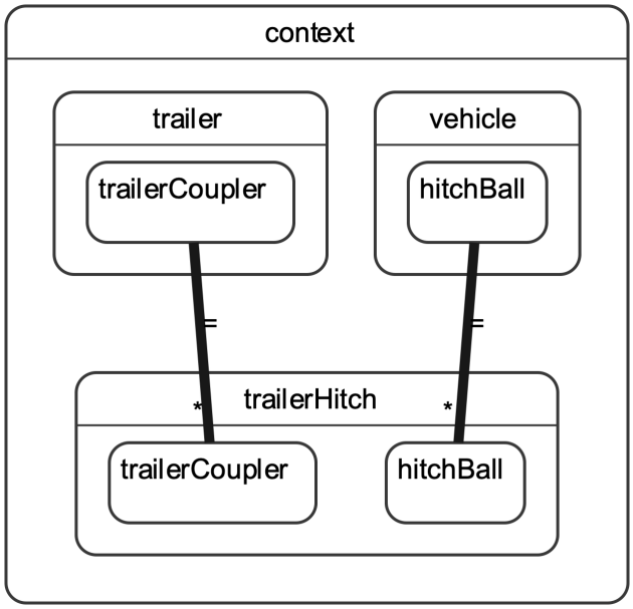


Figure 5a. SysML v2: The vehicle and trailer have reference parts that are bound to the hitchball and trailer coupler respectively

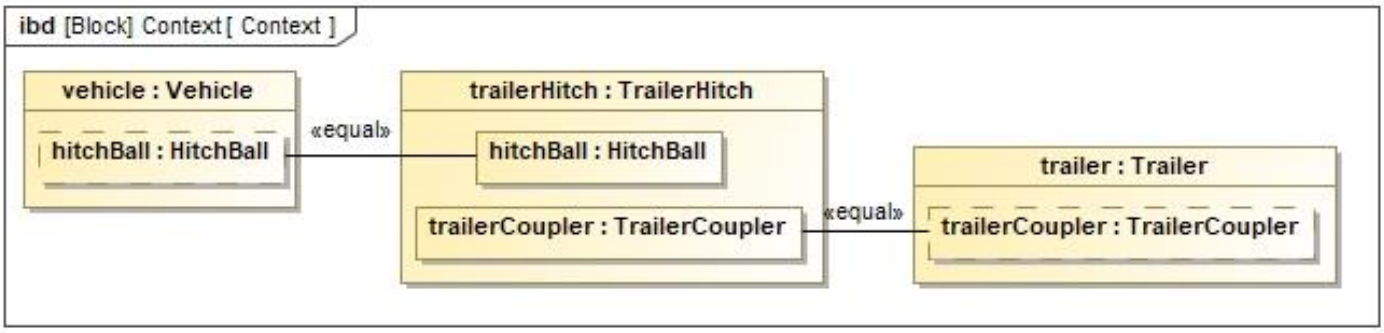


Figure 5b. SysML v1: The equivalent internal block diagram showing the vehicle and trailer with reference parts that are bound to the hitchball and trailer coupler respectively

SysML v1 uses different kinds of relationships to relate its type to its features. As noted above, in SysML v1, a block is generally related to its parts through an association, while a block can contain a port without an association. In addition, a block can contain a classifier behavior such as a state machine or activity. In SysML v2, a part or part definition contains analogous parts, ports, and behaviors as well as many other kinds of features that are all related to its owner using kinds of composite or referential feature memberships. This coupled with the consistent definition and usage pattern described above provides for a more consistent approach to modeling a system in SysML v2.

Variability modeling

SysML v1 provides a capability to model variability that includes the use of specialization and multiplicities. The example in Figure 6a shows a *VehicleFamily* that is modeled as a superset of potential valid configurations that includes an *Engine* with specializations for a 4 cylinder engine and a 6 cylinder engine, a *Transmission* with specializations for a manual transmission and an automatic transmission, and an optional *Sunroof* with multiplicity 0..1. The *Engine* is composed of 4..6 cylinders

with specializations of *Cylinder* for a small diameter and large diameter. The other parts of the *VehicleFamily* include the *driveshaft*, *frontAxleAssembly*, and *rearAxleAssembly* which do not include specializations.

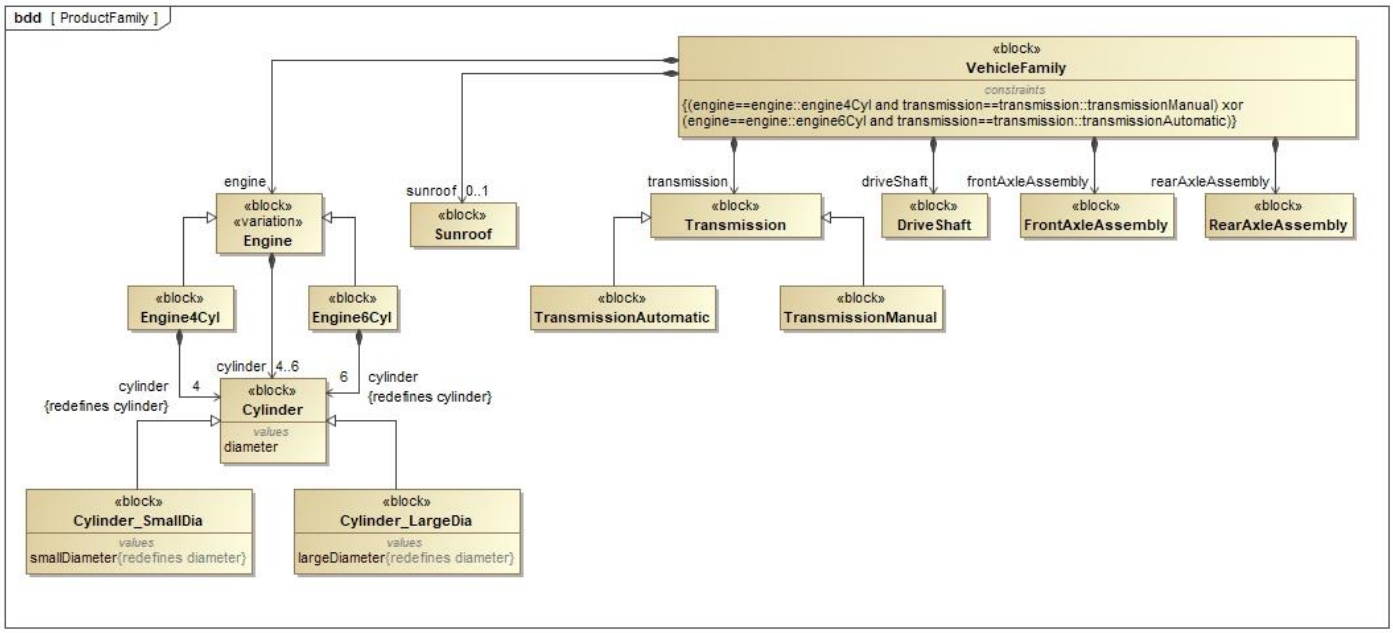


Figure 6a. SysML v1: Model of a vehicle family

The *VehicleFamily* contains a constraint such that if a 4 cylinder engine is selected, then a manual transmission must be selected to be a valid configuration, and similarly, if a 6 cylinder engine is selected, then an automatic transmission must be selected to be a valid configuration.

A particular variant configuration can be defined by specializing the vehicle family and redefining each of the parts to reflect design choices. Two variant configurations are shown in Figure 6b including a *Vehicle_Economy* that has an *engine_4Cyl* and no *sunroof* (i.e., multiplicity [0]), and a *Vehicle_Luxury* that has an *engine_6cyl* with large diameter cylinders and a *sunroof* (i.e., multiplicity [1]). The constraints can be evaluated to confirm that the configurations are valid which means that the *Vehicle_Economy* must have a manual transmission and the *Vehicle_Luxury* must have an automatic transmission. Both configurations inherit the *driveshaft*, *frontAxleAssembly*, and *rearAxleAssembly*.

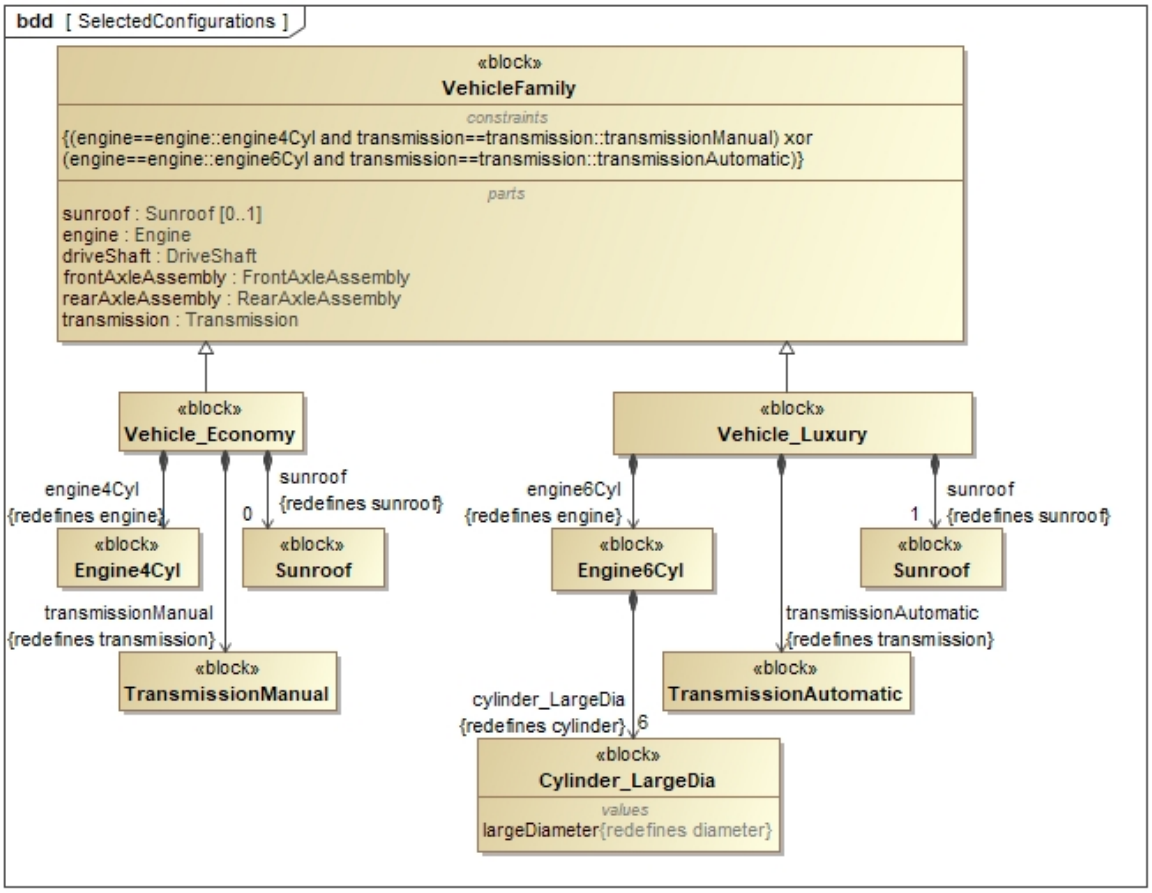


Figure 6b. SysML v1: Variant configurations for Vehicle_Economy and Vehicle_Luxury

SysML v2 introduces explicit concepts to support variability modeling including the concept of variation and variant. A variation can be used to identify any usage element such as an attribute, part, port, action, or requirement as a point of variation. Similarly, a variant identifies a particular choice at a variation point. A selection at one variation point can constrain a selection at another variation point using a standard SysML v2 constraint. In a similar way as described for SysML v1, the superset model of a *vehicleFamily* is shown in Figure 6c. This model includes the set of variations that can be nested to any level. A variant configuration is created by specializing the superset model and selecting a variant for each variation subject to the variability constraints. The equivalent selected variant configurations for the *vehicle_Economy* and *vehicle_Luxury* are shown in Figure 6d.

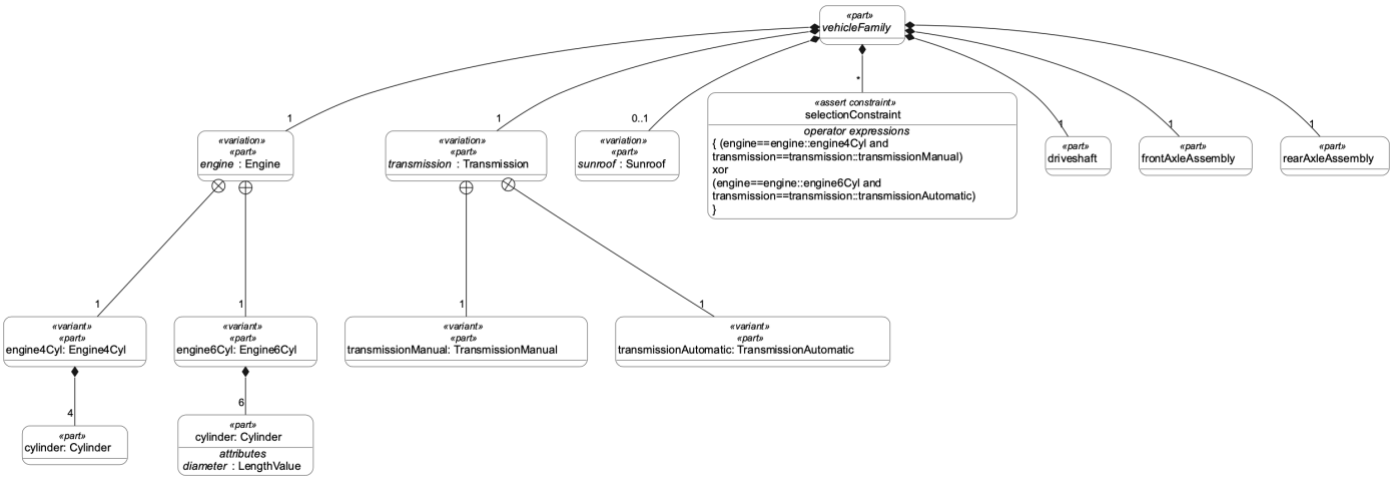


Figure 6c. SysML v2: Model of a vehicle family

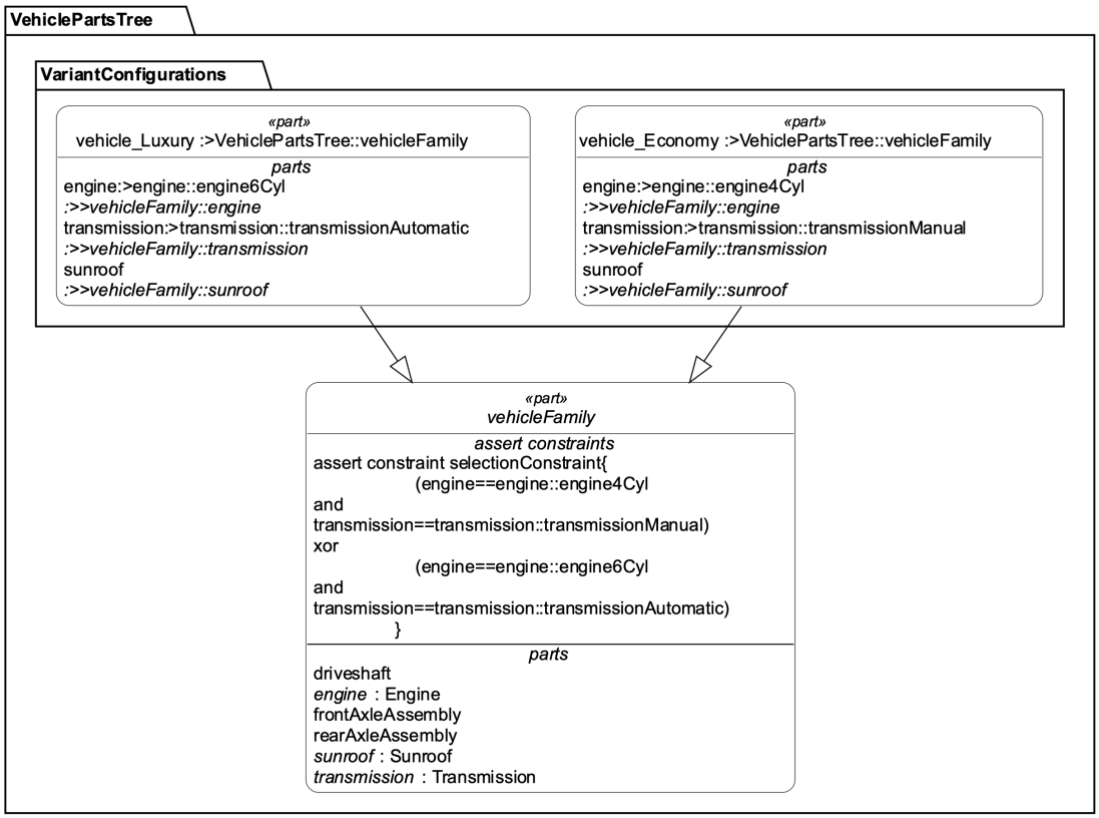


Figure 6d. SysML v2: Variant configurations for Vehicle_Economy and Vehicle_Luxury

The explicit concepts for variation and variant along with the ability to formally specify variability constraints provides a base capability for variability modeling that extends the expressiveness of SysML v2 over SysML v1.

Individuals and snapshots

In SysML v1, an instance specification is used to represent a particular instance of a classifier such as a block. For example, consider the block *Vehicle* in Figure 7a that has a value property *vin* that is typed by *Integer*, and a value property *mass* typed by the value type *kilogram*, and a value property *speed* typed by the value type *km/h*. The vehicle identification number (*vin*) is used to uniquely identify a particular *Vehicle*. An instance of the block *Vehicle* called *vehicle_1* can have the values *vin=1* and

mass=2000 kg. Additional instances of *Vehicle* must be created for each point in time. In Figure 7a, the instances *vehicle_1_t0* and *vehicle_1_t1* are added for times *t=0* and *t=5* along with their corresponding values of *speed*.

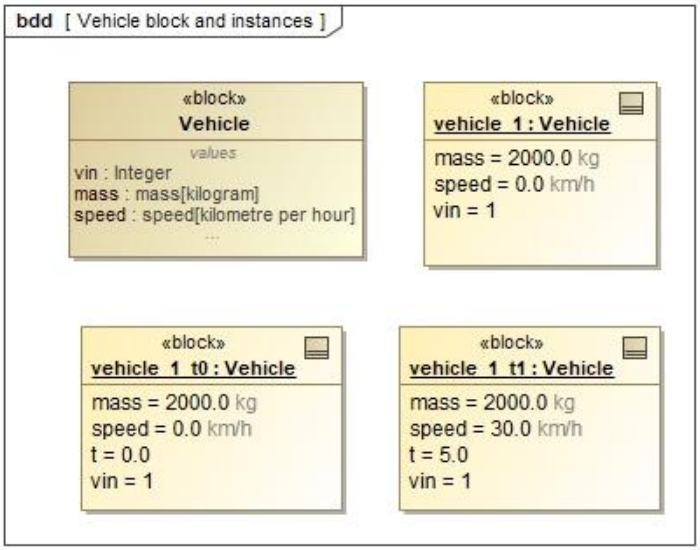


Figure 7a. SysML v1: Vehicle block and instances

SysML v2 introduces the concept of individual and snapshot. An individual represents a particular definition element with a unique identity. For example, *Vehicle_1* can be defined as an individual of the part definition *Vehicle*. Each individual has their own unique lifetime. Different points in its lifetime can be represented as snapshots. For the example above, the *vin* and *mass* are attributes of *Vehicle_1* that are unchanged across its lifetime but *speed* is an attribute of *Vehicle_1* whose value changes across its lifetime. *Vehicle_1* can contain snapshots corresponding to different times in its lifetime that contain specific values of the speed of the vehicle.

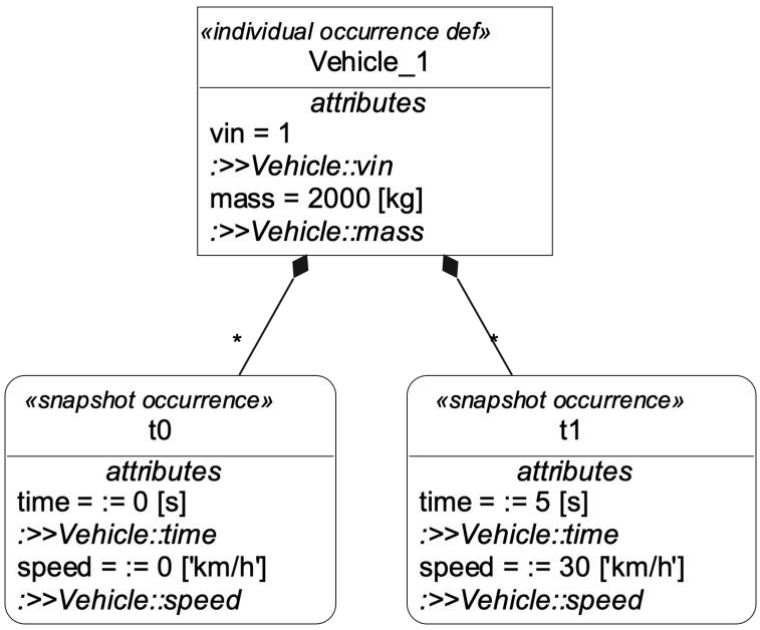


Figure 7b. SysML v2: Vehicle individual and snapshots

SysML v2 clearly delineates the concept of unique identity of an individual from the snapshots that represent different points in the individual's lifetime. As a result, SysML v2 makes it easier than SysML v1 to express concepts such as *Vehicle_1* and the state of *Vehicle_1* as it changes across its lifetime.

4D semantics

SysML v1 is supported by execution semantics that are specified as part of the precise semantics of UML which describe how instances change over time. SysML v2 is grounded in a precise declarative semantics that specify assertions that must be true and correspond to an interpretation of the real world. The semantics are specified in terms of first order logic and are heavily based on the concepts of classification (e.g., classifiers, features, and specialization).

The semantics also give meaning to temporal and spatial concepts. The general concept of an occurrence has temporal extent which is defined by the lifetime of an occurrence. All structural and behavioral entities including items, parts, and actions are kinds of occurrences. Portions of a lifetime are specified as time slices, and a snapshot is a time slice with zero-time duration. The semantics of succession assert that one portion of a lifetime occurs after another. SysML v2 also provides the semantics to quantify time using a default or user specified clock.

A spatial item is a kind of occurrence with spatial extent. The semantics enables spatial items to have a one-, two-, or three-dimensional shape that can be positioned and oriented relative to a coordinate frame. The semantics give meaning to concepts such as inside, outside, or on the boundary of a spatial item. Topological operators provide the ability to define unions, intersections, and subtractions of shapes, and vector math can be applied to define coordinate transformations.

The semantics provide expressiveness and precision that enable sophisticated analysis to be performed on SysML v2 models well beyond what is possible in SysML v1.

Language extension

SysML v1 provides a language extension capability that includes profiles with stereotypes to represent domain specific concepts. SysML v1 enables a stereotype to be defined as an extension of an existing SysML v1 concept. Stereotype properties and constraints can be added to the stereotype. For example, a stereotype «hardware» can be defined that extends block. Stereotype properties for the hardware *manufacturer* and *age* can be added. A constraint can be included to restrict the age of the hardware so that its manufactured date must be less than 10 years. This stereotype can then be applied to any block to represent a hardware item. The manufacturer, age, and the constraint can be specified in natural language or in another language so that the constraint can be evaluated.

A block for a generic *hardwareItem* can also be defined which constrains the characteristics of any *hardwareItem* such as adding a value property for mass to this block. The stereotype hardware can be applied to this block.

The example in Figure 8a shows a SysML v1 block definition diagram of a *Control Subsystem* that is composed of a hardware *Processor* and software *Controller*, and the *Controller* is allocated to the *Processor*. The hardware extension is applied to the *Processor* and the *software* stereotype is applied to the *Controller*.

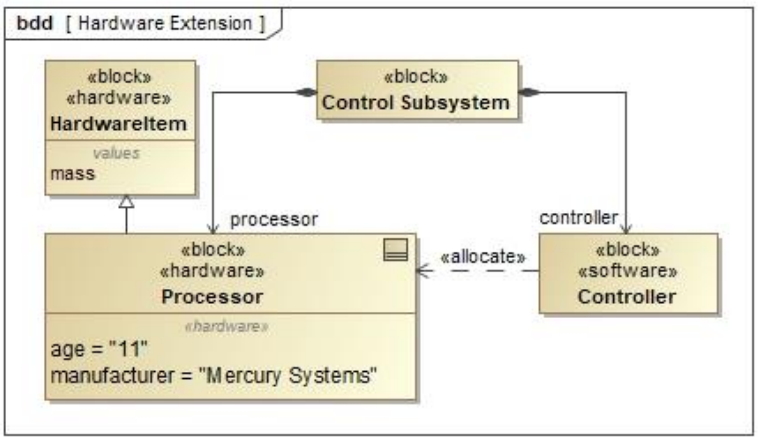


Figure 8a. SysML v1: Applying a language extension for hardware and software

SysML v2 provides an extension capability that provides additional capabilities for representing domain specific concepts. An extension can be defined using metadata that is analogous to defining a SysML v1 stereotype with stereotype properties and constraints. The constraint can be expressed using the SysML v2 expression language.

Figure 8b shows the hardware extension in SysML v2 that corresponds to the SysML v1 extension in Figure 8a. The metadata extension for *hardware* is analagous to the stereotype *hardware* in the SysML v1 example and includes metadata properties for *manufacturer* and *age*. The constraint restricts the age of the hardware to be less than 10 years. The constraint can be expressed directly in SysML v2 and can be a model-level evaluable expression that a conformant tool can check. The SysML v2 *hardwareItem* also includes a restriction that the *subparts* of a *hardwareItem* must also be *hardwareItems*, which would not be possible to specify directly in the SysML v1 model. In SysML v1, an additional constraint must be imposed to ensure that the *hardware* stereotype is applied to any subclass of *hardwareItem*. This is accomplished automatically in SysML v2.

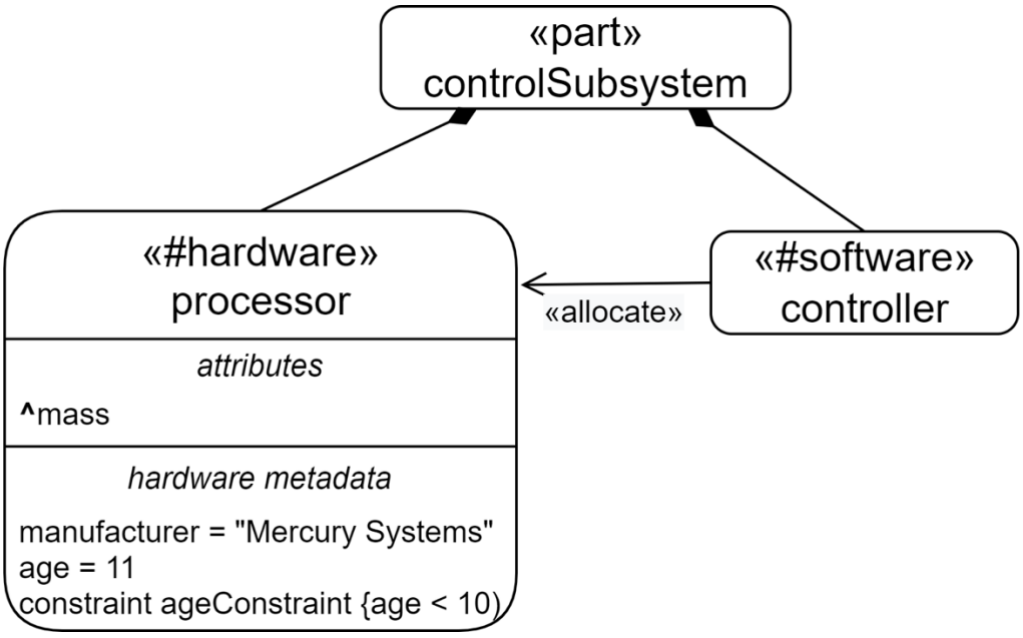


Figure 8b. SysML v2: Applying a language extension for hardware and software

Short name and alias

SysML v2 provides a general capability for providing a short name to augment the name of any element. The short name is included in brackets prior to the name and can be used to provide a numerical identifier. An example is a short name for a mass requirement such as the following:

<'3.2.3.1'> massRequirement.

Any element can also have any number of aliases that apply to a particular namespace. An example is a Sport Utility Vehicle with the alias SUV as shown in Figure 9. The figure also shows a part defined by the alias SUV with a short name <1.1>.

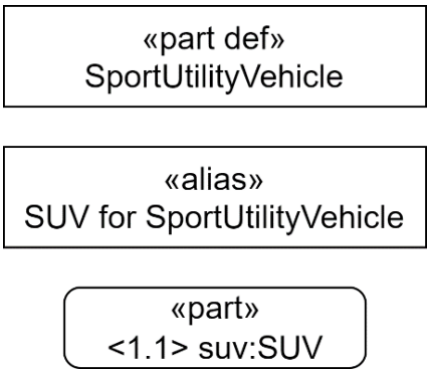


Figure 9. SysML v2: Use of a short name <1.1> and alias SUV

This SysML v2 capability enhances expressiveness and usability over SysML v1 which does not include the ability to define to a short name or alias for any element.

Specific changes in SysML v2 language

In addition to the changes above that apply generally to the language, the following section describes some of the changes in SysML v2 relative to SysML v1 that apply to specific concepts within the language.

Port

In SysML v1, a port can be a proxy port typed by an interface block or a full port typed by a block, or a UML port typed by an interface. In SysML v2, the language was simplified to have a single concept of a port usage defined by a port definition. The port usage and port definition align with the SysML v1 proxy port and interface block. The equivalent of a SysML v1 full port can be represented as a part on the boundary that can connect to other parts external to its owning part. SysML v2 can represent the equivalent of a UML port by including actions as directed features of the port in place of the required and provided operations on UML ports.

A SysML v2 port can have directed features such as data, liquid, or heat that flows into or out of a port. An input or output of an action can be bound to a directed feature of a port to eliminate ambiguity as to which port it flows in or out of. The port can also contain features that are not directed such as voltage. The features can be constrained and evaluated, such as the interface constraints on across and through variables of connected ports. Like any other features, ports can be decomposed and specialized. The simplified port concept in SysML v2 significantly improves usability and precision over the SysML v1 port concepts.

Quantities and units

SysML v1 introduced a comprehensive library called Quantities, Units, Dimensions and Values (QUDV) to specify quantities and units consistent with the international metrology standards. This library enabled quantity and units to be associated with a value type. For example, a value type can represent a real number for the quantity *length* and the unit *meter*.

SysML v2 further improved this library to make it more precise, usable and expressive. In SysML v1, a different value type is required for each unit. For example, a different value type is required to type a value property in meters and a value property in feet. In SysML v2, the attribute definition analogous to a value type is associated with the quantity and not the unit. For example, an attribute defined by an attribute definition for the quantity *length* can be represented as length in meters or feet. This makes quantities and units much easier to use.

The SysML v2 textual syntax can be used to perform quantity arithmetic to create new quantities and units from more primitive quantities and units. An example is the quantity for fuel consumption which can be derived from the more primitive quantities of length / volume and then expressed in the unit of miles per gallon or liters per kilometer (if the quantity is inverted). SysML v2 also enables dimensional analysis to check the quantities and units for compatibility. The expressiveness has been further increased by adding US Customary Units to the standard library.

Requirement

A requirement in SysML v1 includes an id and a text statement to represent a shall statement such as the *MassRequirement* and *VehicleMassRequirement* shown in Figure 10a. A requirement can be extended to include additional attributes. A non-normative extension was added in SysML v1.5 to support property-based requirements which enables a requirement to be augmented by parametric constraints. This is intended to provide the ability to express the requirement with a parametric constraint in addition to a text statement.

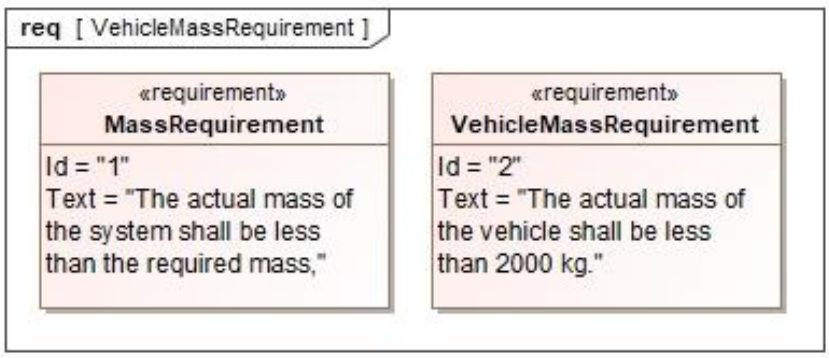


Figure 10a. SysML v1: A Mass Requirement and Vehicle Mass Requirement

A requirement in SysML v2 is a kind of constraint. In addition to the id and text statement, a SysML v2 requirement can leverage the SysML v2 expression language to formally express a constraint expression that can be evaluated as true or false. For example, the constraint for a mass requirement can be expressed as *{massActual <= massRequired}*. The evaluation of the constraint can compare the actual mass from either an analysis or a test and determine whether the requirement is satisfied. In addition, a SysML v2 requirement can include assumed constraints which must be evaluated to true for the requirement to be applicable. The requirement follows the definition and usage pattern so

that a definition can specify a generic requirement and a usage can be specific to its context. An example of a *massRequirement* that is defined by a more general *MassRequirement* is shown in Figure 10b. The *massRequirement* includes an assumption that the *fuelMass* <= 40 kilgorams. The SysML v2 requirement is more precise, expressive and usable than a SysML v1 requirement.

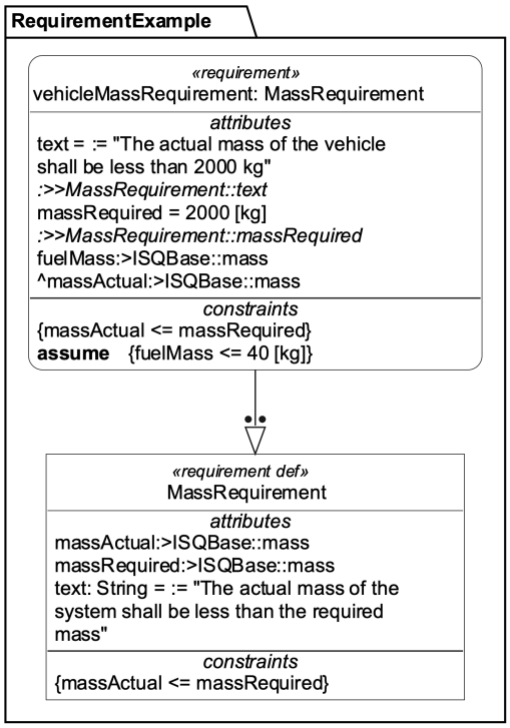


Figure 10b. SysML v2: A Vehicle Mass Requirement defined by a Mass Requirement

Annotations

SysML v1 includes the concept of a comment that can be used to annotate any kind of element. A single comment can annotate multiple elements and any element can be annotated by multiple comments.

SysML v2 provides significant improvements in the ability to annotate elements. A SysML v2 comment is a kind of annotating element that can annotate any other element. Unlike SysML v1, the annotation relationship between the annotating element and the annotated element is an actual relationship. This enables annotations to be navigated like any other relationship.

A comment is an annotating element that provides additional information about the element it is annotating. In SysML v2, this concept is further elaborated by defining metadata as a special kind of annotating element that can have additional structure. A metadata definition is a kind of attribute definition that can contain additional attributes with values. This enables structured metadata to annotate any kind of element.

SysML v2 also includes standard library extensions of metadata for risk, status, and other kinds of annotating information. The concept of metadata is also used to define the language extension mechanism that correspond to a SysML v1 stereotype as described previously.

A textual representation is another kind of annotating element that includes a language and body. A textual representation can annotate any element like any other annotation. For example, a textual representation can annotate an action with an expression that is defined in another language, similar to a SysML v1 opaque action. The SysML v2 annotating elements provide consistency and expressivity

well beyond the capability of a SysML v1 comment.

Cases including use cases, analysis cases, and verification cases

SysML v1 includes concepts for use case and test case. A use case specifies a behavior by defining how the subject interacts with external entities called actors. A test case specifies how a requirement is satisfied.

SysML v2 includes a general concept of a case that contains a subject, actors, an objective, and a set of actions that are used to satisfy the objective and return a result. This general concept is specialized for use cases, analysis cases, and verification cases. This provides a consistent way to represent these concepts as well as a basis for further extensions such as for specifying assurance cases and safety cases.

The use case provides a way to specify interactions between a system or enterprise and external entities. An analysis case describes the actions needed to perform an analysis such as a set of calculations. A special kind of analysis case called a trade study is in the standard SysML v2 library. The analysis case adds capabilities beyond what a SysML v1 parametric diagram can do to specify an analysis. A verification case describes the steps needed to verify that a system or other element satisfies its requirement and provides a more general capability than a SysML v1 test case, such as a standard way to specify verification objectives.

Other differences

SysML v2 includes many other concepts that increase the precision, expressiveness, consistency, and usability relative to SysML v1 which can be the subject of future publications. Examples include enumeration, item, n-ary dependency, allocation, cause-effect relationship, filter expression, and others.

Interoperability

SysML v1 includes the XML Metadata Interchange (XMI) standard as a means for serializing a model and interchanging the serialized model file with other tools. In addition, SysML v1 tool vendors typically provide a custom application programming interface (API) to provide access to their SysML v1 models.

SysML v2 significantly increases interoperability beyond SysML v1 by providing a more robust model interchange standard and a standard API to access the systems model from a conformant SysML v2 tool. In addition to XMI as a standard for model interchange, SysML v2 also includes a JSON representation which is consistent with the commonly used REST interface that many modern tools support. The textual syntax also provides a means for model interchange subject to the limitation that the element id is not directly available.

As noted in an earlier section, SysML v2 includes a standard API and associated standard set of services to navigate, query, and update the model. This capability enables interoperability with other tools and software applications and improves the ability to leverage the system model as part of the broader digital engineering environment. Other applications can access the SysML v2 model in a standard way and operate on this data, and then return results to the SysML v2 modeling tool to further update the system model.

The Systems Modeling API & Services specification also includes standard recipes for performing higher level procedures that support systems engineering activities. Examples include a standard way to return a parts decomposition, action decomposition, or requirement decomposition. The consistency of the SysML v2 language facilitates the development and maintenance of additional

standard procedures to support other systems engineering activities such as change impact assessments.

Summary

SysML v2 was developed to address many of the limitations of SysML v1 by improving the precision expressiveness, consistency, usability, extensibility, and interoperability of the language. The result is

a language that can improve adoption and effectiveness of MBSE with SysML that will scale with increasing system complexity.

SysML v2 was rearchitected and redesigned to meet the requirements in the SysML v2 RFP. It includes a new metamodel that is not based on UML, the introduction of a textual syntax in addition to the graphical syntax and a more flexible view and viewpoint mechanism for presenting the model information to stakeholders. SysML v2 also includes a new standard API that enables interoperability with other engineering models and tools.

There are many differences between SysML v2 and SysML v1 that are of a general nature and apply across the SysML v2 language. This includes the consistent definition and usage pattern, usage-based decomposition, a reified membership relationship, variability modeling, individuals and snapshots, and 4D semantics. In addition, there are many differences that relate to specific modeling concepts such as ports, quantities and units, requirements, cases, annotations, and others. This paper highlights how these differences support the language objectives.

The SysML v2 specification is entering the finalization phase where tool vendors can raise implementation issues they encounter as they develop their SysML v2 tools. The SysML v2 specification is expected to become a final adopted specification in 2024, and it is anticipated that some of the tool vendors will have tools available around this time.

List of Acronyms Used in this Paper

Acronym	Explanation
API	Application Program Interface
BNF	Backus Normal Form
INCOSE	International Council on Systems Engineering
KerML	Kernel Modeling Language
MBSE	Model-based Systems Engineering
OMG	Object Management Group
RFP	Request for Proposal
RTF	Revision Task Force
SST	SysML v2 Submission Team
SysML	Systems Modelling Language

Acknowledgements

The authors acknowledge the contributions of the SysML v2 Submission Team (SST) members. There are over 80 organizations that identified representatives to participate in the SST, and they are listed in the acknowledgement sections of the SysML v2 final submission (references 5, 6, and 7). Many

representatives have contributed to this effort; they are too numerous to mention. A special acknowledgement goes to the other current and previous SST track leads that provided the day-to-day leadership for this effort. These individuals include Manas Bajaj, Yves Bernard, Bjorn Cole, Charles Galey, Karen Ryan, and Tim Weilkiens. In addition, the authors acknowledge and appreciate Dassault Systèmes for the use of Cameo Systems Modeler™ to represent the SysML v1 models.

References

1. Friedenthal, Sanford. *Requirements for the Next Generation Systems Modeling Language (SysML® v2)*, INCOSE INSIGHT, March 2018, Volume 21, Issue 1, pages 21-25.
2. *Object Management Group Systems Modeling Language (OMG SysML™) Version 1.7,* OMG Document: ptc/22-08-02, December 2022.
3. *Object Management Group Systems Modeling Language (SysML®) v2 Request For Proposal (RFP),* OMG Document: ad/2017-12-02, 09 December 2017.
4. *Object Management Group Systems Modeling Language (SysML®) v2 API and Services Request For Proposal (RFP),* OMG Document: ad/2018-06-03, 21 May 2018.
5. *Object Management Group Kernel Modeling Language Version 1.0 Final Submission,* OMG Document: ad/2023-03-02, March 2023.
6. *Object Management Group Systems Modeling Language (SysML®) Version 2.0 Final Submission Parts 1, 2*, OMG Documents: ad/2023-03-03, ad/2023-03-06, March 2023.
7. *Object Management Group Systems Modeling Application Programming Interface (API) and Services Version 1.0 Final Submission,* OMG Document: ad/2023-03-08, March 2023.
8. Seidewitz, Ed, *On a Metasemantic Protocol for Modeling Language Extension*, MODELSWARD 2020, Valletta, Malta, February 2020.
9. Friedenthal, S., Seidewitz, E., *A Preview of the Next Generation System Modeling Language (SysML v2)*, Project Performance international Systems Engineering Newsjournal, PPI SyEN 95 – 27 November 2020.

About the Authors

Sanford Friedenthal

Mr. Friedenthal is an industry leader and independent consultant in model-based systems engineering (MBSE). He has over 30 years of experience as a system engineering practitioner, department manager, and leader of organizational initiatives for Lockheed Martin. Mr. Friedenthal also has been a leader of the industry standards effort through the Object Management Group (OMG) and INCOSE to develop the Systems Modeling Language (OMG SysML®) that was adopted in 2006. He is co-author of “A Practical Guide to SysML” and “Architecting Spacecraft with SysML” and co-led the SysML v2 Submission team to develop the next generation of SysML v2.

FEATURE ARTICLE

Ed Seidewitz

Ed Seidewitz is Chief Technology Officer at Model Driven Solutions, Inc., a long-time provider of enterprise and systems architecture services using model-based methods. Mr. Seidewitz has over 30 years of professional experience with the modeling, architecture, and development of systems spanning diverse domains including aerospace, finance, acquisition, and health care. Mr. Seidewitz has been active with the Object Management Group (OMG) for 20 years, including involvement in every UML Revision Task Force and with the Service Oriented Architecture Modeling Language (SoaML) and System Modeling Language (SysML) specifications. He was primary author of the Foundational Subset for Executable UML Models (fUML) and Action Language for Foundational UML (Alf) specifications and maintains open-source reference implementations for both. He was also chair of the OMG Model Interchange Special Interest Group and the Revision Task Forces for fUML, Alf, Precise Semantics of UML Composite Structures, and Precise Semantics of State Machines, and he co-led the SysML v2 Submission Team.

SYSTEMS ENGINEERING RESOURCES

Useful artifacts to improve your SE effectiveness

NAFEMS Blog: Model Calibration and Model Validation - What's the Difference?

NAFEMS, the International Association for the Engineering Modelling, Analysis and Simulation community, maintains an active and informative blog.

Check out the recent post on Model Calibration and Model Validation – What's the Difference? *In the post, Gregory Westwater and William Oberkampf, members of NAFEMS'* Simulation Governance and Management Working Group (SGCWG) *distinguish the two concepts and discuss where each is most useful. View the discussion video* here.

Other posts from the first quarter of 2023 include:

- The Case for Reckless Engineering
- Is your modelling and simulation capability up to scratch?

Subscribe to the NAFEMS blog here.

NAFEMS E-Learning Courses for May – June

NAFEMS offers a diverse range of code-independent, in-depth online training courses on engineering analysis and simulation. Upcoming courses beginning in May - June 2023 include:

Simulation of Lubricated Contacts (4 sessions; 2 May – 5 June)
This course is designed to develop skills and knowledge in simulating lubricated contacts using numerical methods. Lubricated contacts form a lubricant film which completely or partially separates contacting surfaces.

Introduction to Practical CFD (3 sessions; 12 May – 2 June)
This course offers attendees the fundamental knowledge for using Computational Fluid Dynamics (CFD) in real-life engineering applications. Through a simple and moderately technical approach, this course describes the steps in the CFD process and provides benefits and issues for using CFD analysis in understanding of complicated flow phenomena and its use in the design process.

Introduction to Dynamics using FEA (6 sessions; 16 May – 20 June)
This 6-session, live online course will cover a range of topics, all aimed at structural designers and engineers who are moving into the area of dynamic analysis.

Metals Material Modelling: Plasticity (3 sessions; 17-31 May)
The course covers plasticity theories that are widely used to analyze practical engineering applications in metals. Mathematical formulations and equations are intentionally kept to a minimum. Emphasis will be placed on how engineering design incorporates these theories and how the FE method models plasticity. Difficulties encountered by both the FE user and the FE software in modelling plasticity will be highlighted using many examples to demonstrate plastic behavior and how to assess the accuracy

of the FE solutions.

Fundamentals of Multibody Dynamics Simulation (3 sessions; 6-20 June)
This 3-session, live online course explains what you need to know to perform Multibody Simulation Analysis of systems and mechanisms. you will learn about the theoretical, numerical, and methodological background which will allow you to build your first Multibody Dynamic (MBD) models, and then progress to more complex ones. Examples are discussed in detail allowing you to understand not only model construction but how and why a given model has been built that way. The examples used will help illustrate the use of multiple commercial MBD software packages as concrete examples of applications of existing commercial technology.

Understanding Solid Mechanics: Stress Analysis Approaches (5 sessions; 8 June – 6 July)
This course is your starting point towards really understanding solid mechanics. You need to understand the basics to enable you to apply good practice in your finite element analysis, and this course will give you the knowledge you really need for a good understanding of the principles that every engineer or designer should know.

Elements of Turbulence Modeling (2 sessions; 9-16 June)
This 2-session, live online course will cover a range of topics including:

- Understanding turbulence, energy cascade & vortex stretching
- Turbulence scales, time averaging and closure problems
- Boussinesq hypothesis
- Various RANS-based models
- Wall treatment
- y+, Detached Eddy and hybrid models

Composite Finite Element Analysis (5 sessions; 15 June – 13 July)
This 5-session, live online course will cover a range of topics, all aimed at structural designers and engineers. The objective of this course is to break down the composite analysis process into clearly defined steps, give an overview of the physics involved and show how to successfully implement practical solutions using Finite Element Analysis.

Next Steps with Multibody Dynamics Simulation (3 sessions; 27 June – 11 July)
This 3-session course offers guidance on how to assess and plan the task of carrying out advanced Multibody Simulation Analysis of systems and mechanisms. By attending, you will build a theoretical, numerical and methodological background which will allow you to build advanced MBD models.

View the current course listing.
Join the NAFEMS E-Learning Wait List to be notified when a future course of interest will be offered.

Agile and Requirements and Systems Engineering (RE/SE)

The relationship between Agile development methods and systems engineering disciplines, including requirements engineering (RE) and business analysis (BA), continues to be a topic of much interest. Two recent resources address this concern.

Why Your Agile Organization Needs A High-Performing Requirements Engineering Competency
A March 2023 post by Howard Podeswa in the Requirements Engineering Magazine sought to address the future of the RE and BA competencies and roles in organizations that are committed to use Agile development for both software and systems. Podeswa provided data that demonstrate that RE/BA is alive and well in many such organizations and highlighted the value-added by RE disciplines. The

value impacted by RE can be broken into five dimensions:

- User Value
- Business Value
- Learning Value
- Risk Reduction and Opportunity Enablement (RR & OE)
- Time Criticality

Podeswa defines the practice of *Agile RE* and describes numerous RE techniques by which such value is delivered:

- Lean Startup Minimum Viable Product (MVP)
- Minimum Marketable Product (MMP), Minimum Marketable Feature (MMF)
- User Story-Splitting
- Feature Preparation (aka Backlog Refinement)
- Triad Meetings
- Acceptance Test-Driven Development/ Behavior-Driven Development (ATDD/BDD)

Check out other articles from the Requirements Engineering Magazine here.
Subscribe to the Requirements Engineering Magazine.

Agile and Systems Engineering: The Good, The Bad and The Ugly
This keynote presentation by Stuart Jobbins won the Best Paper Award at the INCOSE UK ASEC 2022 conference.

Abstract: Acceptance of Agile approaches is not without controversy, even within software circles, but increasingly the author is asked for his opinion on its application to Systems Engineering. Early Agile texts suggest an "all-or-nothing" mantra that decries other methods as mediocre by comparison, and that 'partial compliance' is invalid. In this paper we consider the major differences in context between Agile's original intended environment of software development and the implications when applied in Systems Engineering. We also consider the implications on the Systems Engineering lifecycle stages of using Agile concepts. We conclude by identifying which components fit well with Systems Engineering, and those that should be avoided.

"

High achievement always takes place in the framework of high expectation.

Charles F. Kettering

FINAL THOUGHTS FROM SYENNA

My final thoughts for this edition have been shaped by the two outstanding articles provided by truly senior thought leaders in the systems engineering community.

In the Forum section, Randall Iliff has proposed a simple three-element model of barriers (Ignorance, Misunderstanding and Ego) to the adoption of all the goodness that SE has to offer. Randall has invited you to contribute to that discussion through a quick online survey based on your experience (perhaps painful) of crashing into these and other (yet to be named) barriers. His hope is that the collective wisdom of SE practitioners will lead to better understanding of this system of barriers and to creative ideas on how to counter them. The motivation for this piece is the "go wide" realization that millions of engineers and other creative problem-solvers on planet Earth could benefit from even a small dose of SE goodness and that small dose, applied at scale, could have huge positive impacts on well-recognized (and likely intractable without some systems thinking) global challenges.

The feature article by Sandford Friedenthal and Ed Seidewitz summarizes the "diff" between SysML v2 and SysML v1 and highlights the potential benefits of refactoring the modeling language standard. You will not find a more compact and well-written explanation of this important work. This "go deep" treatise highlights modeling language enhancements and simplifications aimed at removing barriers to MBSE adoption. It fearlessly tackles the numerous devils that are in the details of modeling the problem domain and solution space of the increasingly complex challenges being faced by SE practitioners – modeling with sufficient completeness, precision, and flexibility to successfully address these challenges.

Both articles shout that "*SE, done well, is worth the investment!*" They recognize barriers to adoption that have limited the reach and global impact of the SE discipline. But they attack these barriers in very different ways.

It would be unwise to treat the "Go wide (to reach more folks)" and "Go deep (to tackle higher-complexity challenges)" messages as mutually exclusive paths to greater SE global value creation. Thinking about them reminded me of two simple models of reality/causality that might show how the two approaches to improving SE adoption could be complementary. Because this is a Final Thoughts post, not an academic journal article, I'll paraphrase my understanding of these two models and point you to the veritable source, Wikipedia, for deeper insights.

To Syenna, the Hierarchy of Competence model asserts that human beings advance in competence in any capability or skill through four stages:

- Unconscious Incompetence
- Conscious Incompetence
- Conscious Competence
- Unconscious Competence

SE is comprised of many such capabilities and the path to growth for each capability begins with the "Oh my!" moment when individuals or teams realize that they are incompetent in an area that very much matters to their success. That state transition from Unconscious Incompetence to Conscious Incompetence is humbling and gives Ego an initial opportunity to erect barriers to growth. Deny the problem. Blame something or someone else. Cast doubt on the merits of any proposed path to

FINAL THOUGHTS FROM SYENNA

Conscious Competence: "That will never work here". The investment required to transition to the state of Conscious Competence gives pause to the rational mind; the time (cycles of learning) required to gain Unconscious Competence may discourage many from starting the climb. "Let's take an easier and quicker path."

A side note – Syenna once worked with an individual, a retired Army leader, who claimed to be one of the originators of this competence model. Neither his age nor his performance backed up his claim. Unconscious Competence wasn't broadly evident to his co-workers.

To Syenna, the Pareto Principle/Law asserts that 20% of "anythings" have 80% of the impact on overall success in any situation, where "anythings" could be requirements, decisions, risks, solution elements, people, process, technologies, etc. I have used this principle in hundreds of situations to discern the *vital few* from the *trivial many* – facing constraints of time and resources, such prioritization of concerns is essential to success.

In the case of SE capabilities, each of us likely has a different list of those capabilities that make the top 20% cut and fall into the *vital few* classification. And the depth of each skill required to ensure success, i.e., the level of Conscious or Unconscious Competence, would also vary based on our experience.

In a non-engineering example, we all hope that our neurosurgeon or cardiologist have chosen the best set of medical diagnostic and treatment capabilities to master. But we also hope that they have gone deep in their understanding human biology, the structure and operation of the brain or heart, and the relevant medical techniques and tools, many levels beyond what we learned in our first elementary health class.

Syenna has spent much of her career trying to make hard things simple. That's been a never-ending task of continuously refactoring models of reality and learning how to transfer these models more efficiently to others. But selecting which things to simplify and pass along to others seems to be the first-order decision. And that's where it takes a global community to get it right and "right" and "rightly-packaged" is determined not by us, but by the folks who most need our help.

Please answer the Call to Action and participate in the Barriers to SE Adoption survey.

Regards,

Syenna