Eldad,   
Please post this email and the UML for SE RFP to the WG WIki. Thank you.  
  
In our WG telecon today, April 14, we agreed on a path forward to develop the Systems Engineering Concept Model (SECM) as follows:  
  
1. The SECM is intended to help specify requirements for the SysML v2 RFP (i.e. what domain concepts must the language express). There will be other requirements beyond those defined in the SECM, such as visualization requirements.  
  
2. We will establish parallel efforts with a Kernel Team and a Domain Team to evolve the SECM as described below. Charles Galey will lead the Kernel Team and John Watson will lead the Domain Team. WG members are free to participate in one, both, or neither of the teams, and should contact the lead.  
  
3. We will use the original UML for SE RFP (attached) as a starting point for developing the domain concepts in the SECM.  
  
The following provides further background regarding this path forward.  
  
In the previous 3 WG telecons, Chas (Charles Gayley) presented a hybrid approach to develop the SECM that included both a small set of kernel concepts such as classifier and feature, and SE domain specific concepts such as system, subsystem, and component. The intent is to use the kernel concepts as building blocks for defining the domain concepts. Yves felt strongly that we should either reuse the UML metamodel or define the domain concepts only without the kernel concepts. He felt the latter would not produce any value.  
  
Given this situation, we agreed to pursue 2 parallel efforts, one of which focuses on defining the Kernel Concepts, and the other focuses on the Domain Concepts. WG members can choose to participate in one, the other, both, or neither activity. Chas will lead the team to develop the kernel concepts and John Watson agreed to lead the team to develop the domain concepts. John is already leading a team to model the SE Use Cases, and it is anticipated that defining the domain concepts will be complemented by the development of the SE use cases.   
  
The Kernel Team will define the kernel concepts with the intent of providing building blocks for the domain concepts. The Kernel Team will leverage UML and SysML metamodel concepts where practical but not be constrained by them. Dick Welling suggested that the Kernel Team identify a qualifier such as quotations to differentiate the Kernel Concepts from the UML/SysML Concepts to avoid confusion.  
  
We then discussed the starting point for defining the domain concepts in the SECM. We agreed to use the UML for SE RFP as the starting point, and then add, modify, delete from there. A simplified subset of the concepts in the UML for SE RFP are contained in a figure in Appendix A.2 of the RFP. The figure shows the concepts and their dependencies. Each concept includes a definition.   
  
By basing the SECM on the original UML for SE RFP, we can evolve a new version of the requirements that are traceable to the original requirements. The original requirements were vetted amongst a broad group. We can also cross check the SECM with the original powerpoint SECM that Rick captured in a model.  
  
The email trail below reflects discussions following out last WG telecon on March 31 regarding the approach to develop the SECM.  
  
Regards,  
Sandy  
  
-----Original Message-----  
From: BERNARD, Yves [mailto:yves.bernard@airbus.com]   
Sent: Thursday, April 02, 2015 2:26 AM  
To: Sanford Friedenthal; 'Eldad Palachi'; 'Galey, Charles E (312E)'  
Cc: 'Delp, Christopher L (313D)'; 'Welling, Dick'; mbse-roadmap-wg@omg.org  
Subject: RE: SECM Discussion  
  
Sandy,  
  
We need to work with well-defined concepts. That is: those of the domain (which is approach b) and eventually those of UML (which is approach a). Using something else (i.e. "in between"), will means using fuzzy concepts in which everyone can put so that we can think we reach a consensus and see that it is not the case when we start the implementation. I don't think that something in between could be efficient and I will not participate to such an approach which will significantly delay SysML 2.0 without clear benefits.  
  
Yves  
  
>-----Message d'origine-----  
>De : Sanford Friedenthal [mailto:safriedenthal@gmail.com]  
>Envoyé : mercredi 1 avril 2015 20:43  
>À : 'Eldad Palachi'; BERNARD, Yves; 'Galey, Charles E (312E)'  
>Cc : 'Delp, Christopher L (313D)'; 'Welling, Dick'; mbse-roadmap-wg@omg.org  
>Objet : RE: SECM Discussion  
>  
>Yves,  
>The 2 approaches you recommend below are at 2 ends of the spectrum.  
>a) building on our past experience and start from the UML metamodel  
>b) start from a strict domain perspective to build a “domain model”  
>  
>I believe the approach Chas and I recommend is somewhere in between. It  
>leverages the UML metamodel and SysML concepts to help us build the  
>domain concepts, while not overly constraining or complicating the concept  
>model.  
>  
>I believe this approach makes sense because we now have many years of  
>experience as a basis for defining the domain concepts. If we were starting  
>from scratch, I would agree that we should apply your golden rule. However,  
>we have the original SECM as a starting point and have been applying SysML  
>for years.  
>  
>It appears we are not able to get a consensus via email. Perhaps we should set  
>up a call with those who have an interest in the SECM approach to see if we  
>can agree on a path forward. Chas has stepped forward to lead this task, so  
>let's make sure we can find a time he is available.  
>  
>I propose next Monday, April 6 at 11:00 AM ET. Can anyone who those who  
>are interested in attending respond to an invite as soon as practical to  
>determine if we have critical mass.  
>  
>Eldad, can you send out a meeting invite, and see if this time works for those  
>who are interested.  
>  
>Regards,  
>Sandy  
>  
>  
>  
>  
>  
>-----Original Message-----  
>From: Eldad Palachi [mailto:eldad.palachi@il.ibm.com]  
>Sent: Wednesday, April 01, 2015 11:55 AM  
>To: BERNARD, Yves  
>Cc: 'Galey, Charles E (312E)'; 'Delp, Christopher L (313D)'; 'Welling, Dick'; mbse-  
>roadmap-wg@omg.org; Sanford Friedenthal  
>Subject: RE: SECM Discussion  
>  
>  
>In my view the purpose of the SECM is just to help define and clarify the  
>terminology used in the RFP.  
>  
>Like Yves, I also don't see the value of going to the abstraction level of  
>elements and relationships.  
>  
>I also don't want to go into discussions for the RFP if something is  
>possible or not because it may or may not violate some meta relationship in  
>the SECM in one of the abstraction layers and I would like to use concrete  
>terms as much as possible rather than use abstract terms since I believe  
>using abstract terms may potentially be confusing to readers.  
>  
>Eldad  
>  
>  
>  
>  
>  
>From: "BERNARD, Yves" <yves.bernard@airbus.com>  
>To: Sanford Friedenthal <safriedenthal@gmail.com>, "'Welling,  
> Dick'" <dick.welling@boeing.com>  
>Cc: "mbse-roadmap-wg@omg.org" <mbse-roadmap-wg@omg.org>,  
>"'Delp,  
> Christopher L (313D)'" <Christopher.L.Delp@jpl.nasa.gov>,  
> "'Galey, Charles E (312E)'" <Charles.E.Galey@jpl.nasa.gov>  
>Date: 01/04/2015 06:32 PM  
>Subject: RE: SECM Discussion  
>  
>  
>  
>Sandy,  
>  
>As you say there are several ways to deal with it, but all don’t have the  
>same efficiency.  
>  
>To me there are only two acceptable approaches: either building on our past  
>experience and start from the UML metamodel (your option 2) or trying to  
>start from a strict domain perspective to build a “domain model” (which is  
>something like your option 5, I think).  
>  
>Designing “domain models” is part of my job for years. One of the “golden  
>rules” we have for such an activity is to avoid using generalization in a  
>first step and to work with domain’s vocabulary only. In other words, we  
>never start from general concepts but from specific ones, the ones which  
>are used by people for that domain for describing the use cases scoping the  
>domain. It is only when we get a first validated version of entities and  
>relations that we can start thinking on how the model could be optimized  
>using generalization and abstract concepts.  
>  
>Yves  
>  
>De : Sanford Friedenthal [mailto:safriedenthal@gmail.com]  
>Envoyé : mercredi 1 avril 2015 16:50  
>À : BERNARD, Yves; 'Welling, Dick'  
>Cc : mbse-roadmap-wg@omg.org; 'Delp, Christopher L (313D)'; 'Galey, Charles  
>E (312E)'  
>Objet : RE: SECM Discussion  
>  
>Yves, Dick,  
>There clearly are multiple ways in which we can approach the Systems  
>Engineering Concept Model (SECM) such as:  
>1. Use the original SECM as a starting point  
>2. Use the UML metamodel as a starting point  
>3. Abstract the SysML metamodel as a starting point  
>4. Create another SECM which leverages the above  
>5. Other  
>  
>The approach Chas is taking is #4. The intent is to develop the simplest  
>concept model that can expresses diverse set of domain concepts, and is  
>highly extensible. The SECM will be used to specify requirements for SysML  
>v2, and is not intended to define an implementation.  
>  
>The following is a brief summary of the Systems Engineering Concept Model  
>(SECM) background and the approach that Chas presented over the last 3 WG  
>telecons. I encourage our WG to give this approach a chance before we  
>abandon it in favor of an alternative.  
>  
>The SECM v1 was used was used to define the requirements in the original  
>UML for SE RFP that led to the development of SysML v1. The SECM included  
>SE Domain concepts that were captured as an informal class diagram in  
>powerpoint along with a semantic dictionary. We created specific text  
>requirements in the UML for SE RFP that corresponded to the domain  
>concepts  
>that were identified as in scope for SysML. During the SysML development,  
>we established a traceability matrix to show how the concepts were  
>satisfied by the language constructs.  
>  
>The SECM Chas is developing is intended to be used in a similar way to  
>specify requirements for SysML v2. This SECM is being more formally  
>captured in a model compared to the original powerpoint. The intent is to  
>cross check the domain concepts for SECM v2 against the domain concepts in  
>SECM v1 to assess coverage of the original requirements.  
>  
>The SECM v2 is intended to leverage our collective experience from the  
>use of SysML, and to refine, refactor, and add concepts that were not  
>evident in SECM v1. There are other concepts that were not included in the  
>original SECM, but are deemed essential, such as the concepts of definition  
>and usage associated with a typed language, and concepts to support variant  
>modeling.  
>  
>The current approach for the SECM is intended to yield the simplest model  
>that can express the diverse range of domain concepts and be highly  
>extensible. An example is to express the domains concept for System,  
>Component, Hardware, Software, etc. in terms of a more fundamental  
>concept,  
>such as block as we have done in SysML.  
>  
>This approach defines a small set of kernel concepts that are used as a  
>basis for defining the domain concepts. The kernel concepts leverage  
>concepts used in UML and SysML, including element, relationship, classifier  
>and feature, which will be further extended to define concepts such as  
>block, behavior, etc. From these, we can define domain specific concepts  
>that are in the original SECM as well as others.  
>  
>To reemphasize, the intent is to leverage concepts from UML and SysML  
>where  
>it makes sense to define the requirements, but not to use UML metamodel.  
>The UML metamodel is a valid starting point for an implementation but is  
>too complex a starting point for our requirements model.  
>  
>My suggestion is to proceed for the next few telecons with the approach  
>Chas has taken, to give an opportunity to define the kernel concepts and  
>perhaps a few examples of the domain concepts. At that time, we can assess  
>whether we need to shift directions. For example, we can replace the kernel  
>concepts with the corresponding UML metamodel elements (option 2 above),  
>or  
>we can delete all of the kernel concepts and focus on the domain concepts  
>only (option 1 above).  
>  
>I believe the current SECM approach will enable us to address the concepts  
>Yves would like to address below such as ports, protocol, parts, variant,  
>etc., while at the same time addressing concepts from the original SECM,  
>and many others domain concepts. It will give us the flexibility to define  
>the diverse range of domain concepts without making the requirements  
>model  
>overly complex.  
>  
>My request is to give this a chance and help make this approach a success,  
>and if it is not working well, then we can try an alternative.  
>  
>Regards,  
>Sandy  
>  
>  
>  
>From: BERNARD, Yves [mailto:yves.bernard@airbus.com]  
>Sent: Wednesday, April 01, 2015 3:10 AM  
>To: Delp, Christopher L (313D); Welling, Dick; mbse-roadmap-wg@omg.org  
>Subject: RE: SECM Discussion  
>  
>In his message below, Dick broaches number of questions that will be worth  
>discussing at one moment or another for SysML 2.  
>  
>Here are my thoughts about the discussion we had yesterday:  
>  
> · Using the UML metamodel as a starting point for our  
> conceptual model is not working on an implementation but on a  
> representation since there is no restriction in the concept we can  
> model (this would not be the case with a profile, for instance).  
> · Referring on Dick’s question, the conceptual model is  
> clearly at M2 but, since we are not discussing about an  
> implementation of this metamodel, we don’t need to worry about M3  
> modeling languages, at that time at least.  
> · Discussing about so basic concepts like “Element”,  
> “Relationship” or “Constraint” is just reinventing the wheel and – to  
> me – a waste of time and effort. I won’t support it. They are not  
> “domain specific” and there is no reason we get to a different result  
> without missing something. Keep in mind that we have the ability to  
> create new metaclasses if UML’s one appears to be incomplete or not  
> sufficient.  
> · As underlined by Dick our resources could be more usefully  
> spent in discussing concepts like “ports”, “protocols”, “parts”,  
> “variant”, etc…  
>  
>Yves  
>  
>De : Delp, Christopher L (313D) [mailto:Christopher.L.Delp@jpl.nasa.gov]  
>Envoyé : mardi 31 mars 2015 23:51  
>À : Welling, Dick; mbse-roadmap-wg@omg.org  
>Objet : Re: SECM Discussion  
>  
>I am confused. My understanding was that we were only working on a  
>conceptual model right now.  
>  
>Seems like all of this would be decided IF we decide to implement anything.  
>  
>From: <Welling>, Dick <dick.welling@boeing.com>  
>Date: Tuesday, March 31, 2015 at 1:38 PM  
>To: "mbse-roadmap-wg@omg.org" <mbse-roadmap-wg@omg.org>  
>Subject: SECM Discussion  
>  
>I think this would eventually go in JIRA but for now…  
>  
>Some points on today’s “discussion,” which I see as being a classic  
>apple-and-oranges conundrum, in this case being the mixing of M-levels:  
>  
> 1. May we assume that SysML2 development will conform to the  
> four OMG meta-levels M3-M2-M1-M0?  
> 2. May we assume that SysML2 will be a metamodel (M2) and as  
> such needs a meta-metamodel (M3) to express its concepts?  
> 3. May we assume that such a meta-metamodel will implement a  
> type system for strictly defining fundamental syntactic and semantic  
> metamodel concepts?  
> 4. Of course, such a typing meta-metamodel already exists in the  
> form of EMOF (or ECORE). Whatever typing meta-metamodels are  
> available, if we don’t adopt one from the outset we are effectively  
> attempting to invent or reinvent one. For example, where does the  
> SECM Container metaclass lie? Is it part of the SysML2 metamodel (M2)  
> or have we inadvertently extended EMOF (M3), which ought to be out of  
> scope or at least done more precisely? Another example of reinvention  
> is the recommendation for adding an ID, but an identifier property is  
> already an EMOF concept.  
> 5. M3 EMOF is a subset of the UML Kernel, so employing it to  
> author SysML2 should obviate most worries about the otherwise complex  
> M2 UML metamodel complicating SysML2.  
> 6. That said, the top-down concept definition process will  
> always evoke this M-level confusion because of the recursive nature  
> of MOF and UML (assuming we use MOF as the SysML2 metalanguage).  
>Why  
> not get out of the metalanguage invention business (though  
> inadvertent) and begin with existing SysML1 (M2) concepts to see how  
> they can be simplified and then generalized using M3? Not bottom-up,  
> rather this approach would proceed from the known and specific to the  
> general and abstract instead of the other way around.  
> 7. For example, Nicolas Rouquette reports an issue (Key:  
> SYSMLR-148, Legacy Issue Number: 19328) where he identifies about 20  
> syntactic and semantic variations for SysML 1.3 ports. Surely not all  
> these are applicable to real system interfaces (at least not in a way  
> that can be readily understood by most systems engineers), so an  
> opportunity exists to (a) collect real system interfaces, (b) model  
> them using 1.3 constructs, (c) evaluate how well these models satisfy  
> the proposed requirements for model construction, data extraction,  
> visualization, etc., and (d) how these might be expressed in revised  
> metamodel form for further evaluation and optimization.  
>  
>  
>Dick Welling  
>Senior Consultant  
>Boeing Systems Engineering  
>480-891-9532  
>OMG Certified SysML Modeling Professional - Advanced  
>  
>  
>The information in this e-mail is confidential. The contents may not be  
>disclosed or used by anyone other than the addressee. Access to this e-mail  
>by anyone else is unauthorised.  
>If you are not the intended recipient, please notify Airbus immediately and  
>delete this e-mail.  
>Airbus cannot accept any responsibility for the accuracy or completeness of  
>this e-mail as it has been sent over public networks. If you have any  
>concerns over the content of this message or its Accuracy or Integrity,  
>please contact Airbus immediately.  
>All outgoing e-mails from Airbus are checked using regularly updated virus  
>scanning software but you should take whatever measures you deem to be  
>appropriate to ensure that this message and any attachments are virus free.  
>  
>  
>  
>  
>The information in this e-mail is confidential. The contents may not be  
>disclosed or used by anyone other than the addressee. Access to this e-mail  
>by anyone else is unauthorised.  
>If you are not the intended recipient, please notify Airbus immediately and  
>delete this e-mail.  
>Airbus cannot accept any responsibility for the accuracy or completeness of  
>this e-mail as it has been sent over public networks. If you have any  
>concerns over the content of this message or its Accuracy or Integrity,  
>please contact Airbus immediately.  
>All outgoing e-mails from Airbus are checked using regularly updated virus  
>scanning software but you should take whatever measures you deem to be  
>appropriate to ensure that this message and any attachments are virus free.  
>  
>  
>  
>  
>  
>  
>  
>This mail has originated outside your organization, either from an external  
>partner or the Global Internet.  
>Keep this in mind if you answer this message.  
>  
  
  
The information in this e-mail is confidential. The contents may not be disclosed or used by anyone other than the addressee. Access to this e-mail by anyone else is unauthorised.  
If you are not the intended recipient, please notify Airbus immediately and delete this e-mail.  
Airbus cannot accept any responsibility for the accuracy or completeness of this e-mail as it has been sent over public networks. If you have any concerns over the content of this message or its Accuracy or Integrity, please contact Airbus immediately.  
All outgoing e-mails from Airbus are checked using regularly updated virus scanning software but you should take whatever measures you deem to be appropriate to ensure that this message and any attachments are virus free.