Hello Team,

Last time we met, we promised each other  "thoughtful messages".  In that spirit, I'd like to share a few observations regarding the use case data from the workshop.  Note the attached files for reference.  In the first file, I've added page numbers to the use case document for reference.  In the Excel file, I've applied a use case number and a page number to reference the use cases.  Note that the **use case number** = **page number** - 3.  An asterisk denotes that a use case was one of the key use cases identified at the end of the workshop.

Please do forgive me if some of the follow observations seem obvious.  I'm being explicit about these points because many of these things seem to be worthy of further discussion.  Also, if I've missed something important or you have a different view, please do chime in.  Observations:

* There are 35 use cases, if we include the use cases contributed by Robert Karban from the Telescope Challenge Team.
* Although we're calling the content of each of the 35 filled in forms "use cases", most of them are in a partial state of completion and most are somewhat terse (and potentially ambiguous).  An agile programmer would refer to many of them as "user stories" rather than use cases.  By "user story" I mean a promise to talk about a particular use case at some future time, rather than an actual use case.  Virtually every use case that we have would benefit from adding further detail.  This is a natural outcome given that the meeting time at the workshop was so restricted and only 2/3 of the expected meeting time was actually available to the team.  It was rare to see a use case with over 6 sequential tasks listed.  Many listed no tasks at all.
* Although there will naturally be more than one way to categorize the use cases, the attached spreadsheet categorizes the use cases mainly by SE process aspect as:
  + Agreement
  + Requirements
  + Architecture
  + Analysis and Simulation
  + Report Generation
  + Reviews
  + Decision-making
  + Utilities and Tool Capabilities - These are usually at a more detailed level and are potentially used by several other use cases.

Several of the use cases spanned several aspects or could be applied in more than one SE process; however, this complexity is not treated in detail in the analysis.  In such situations the use case was allocated to one process aspect but could be legitimately placed in other categories.

* In many cases, the workshop teams independently identified essentially the same use cases.  Others overlap.  Effectively, of the 35 use cases, approximately 20 are independent.  For instance, if we look at the 4 use cases selected as the most important contributions from the workshop two are virtually identical:
  + Use Case 3 (Page 6) Assemble components from library to meet mission needs
  + Use Case 31 (Page 34) Define System architecture and conduct architectural analysis
  + In fact 4 of the architectural use cases were very similar (3, 10, 12, and 31)

In practice, seeing a use case repeated is a strong indication of its importance.  It's also interesting that the perspective on a repeated use case sometimes differed in interesting ways.  For instance, use case 3 emphasized a library for architecting and use case 10 emphasized using rules.

* There were a number of repeated themes that appeared in many use cases:
  1. Collaboration (Ron mentioned this at our last meeting.) - This is explicit in some use cases but implicit in almost all of them.
  2. Computable Models - Many of the use cases assume SysML models that are computable.  Essentially, it was assumed that describing a model led to a direct calculation of the performance characteristics of the resulting system.
  3. Integration with Analysis and Simulation Tools - Many of the use cases assume integration with modeling and simulation tools.  For instance, the MathWorks tools were mentioned in one of the use cases.
  4. Integration with Discipline-Specific CAD Tools - Several of the use cases assumed integration with discipline-specific CAD tools.

There is much more that could be said about individual use cases.  Please add your thoughts...

In terms of a path forward, I very much like (Ron's suggestion) from the last meeting that we proceed to develop use case examples.  This would have several benefits:

* An example naturally evokes details that are essential for analysis.  The details of process steps reveal which design objects are manipulated and how they are manipulated.
* Examples remove ambiguities from terse use case descriptions.
* Gaps in existing MBSE environments are revealed more clearly.  This benefits stakeholders such as tool vendors.  Some gaps that appear so far include:
  + Collaboration - Support for collaboration capabilities within existing MBSE environments are currently limited to non-existent.  This is a major opportunity for tool vendors.  The exact collaboration functions needed will be more clearly  revealed by detailed use cases.  Priorities for varying collaboration affordances will be more clearly revealed.
  + Computable Models - Existing SysML models are mostly used declaratively.  The priority of computability features will be more clearly revealed by examples.  This will help us and potentially also help the ontology group prioritize its efforts.
  + Integration - Although integration ambitions are vast, current integration within MBSE environments is limited.  Use case examples will highlight the most important gaps.  This is important because integration is hard work and vendors need to prioritize.
* Use case examples will be a concrete and necessary step forward for us.
* It is necessary that we get into the details as part of the process of designing usability experiments.

I suggest that for use case examples, we need:

* Detailed step by step examples, down to approximately the level of the mouse click.
* For each use case example , an example model is that will be manipulated at each step a use case example
* In some cases, where there is a missing function or capability in the tool that is used for the use case example, there will need to be a mockup of the action of the missing function.  Constructing such mockups is an important activity that reveals key details.

Your thoughts?

Cordially,

Scott