1 Mar, 2018

Robert Trzeciak

Program Manager

Northrop Grumman IS

468 Viking Drive

Virginia Beach, VA 23452

757-498-5544 work

757-635-2628 mobile

**Progress Report – 1 Feb 2018 – 28 Feb 2018**

Contract Number: HSHQDC-06-D-00022

Contract Number 7500097279

Order Number: HSCG23-07-J-TED150

Task Order – Performance Work Statement (PWS) 1.12

Attachments: (1) SAROPS subcontractor financial reports.

1. **Lots of meetings. Long meetings.**
2. **Did multi-sensor, which subsumes “other sensor.” A few shallow bugs, but nothing major. Appreciated Rob and Jim finding the bugs.**
3. **Documented the transit xml that should clear up some confusion about the problem. In general, thoroughly understanding the data of a problem, -what each input datum is used for, what its units are, and how it is used to produce an answer, and similarly for the numbers that constitute an answer, is critical for even discussing the problem. I call this “4th grade math” in honor of the story problems that appeared in the last section of each chapter in my 4th grade math book. My document provides much of the 4th grade math, as it describes the xml. In fact, the document is simply an example xml with comments, similar to the one I provided last month after I had thrashed out the details of other sensor.  
    This document describes what a StandAlone PatternVariable’s data are and what a non-StandAlone PatternVariable’s data are. It is basic fundamental knowledge for this problem that StandAlone and non-StandAlone PatternVariables are the two very similar but very fundamentally different parts of a Planner problem. Also, that there is no difference between a GetInitial, an Optimization, and an Eval problem; they all have StandAlone and/or non-StandAlone PatternVariables, and the answer files are essentially the same.  
    I noticed that Jim now uses the term “filter” when describing sensors, and that came from my write-ups and discussions. Similarly, I hope people will use the terms StandAlone and non-StandAlone PatternVariables now.**
4. **Prior to 2.2, we had only StandAlone PatternVariables and, when we have only StandAlones, 2.2 should work as well as 2.1.1. It does. I was testing with a case that had valid StandAlone variables being associated with an invalid PatternVariableSequence. Although I should have thrown the entire PatternVariableSequence away (including its PatternVariables), I processed the PatternVariables as StandAlones. But then I was given a valid PatternVariableSequence and I tried to process its PatternVariables as non-StandAlone PatternVariables. My code is not at that stage yet and it was crashing. Judy agreed not to send me any PatternVariableSequence at all, which is what was meant by testing that 2.2 could run old-style Planner problems; if you give me a new-style Planner problem complete with transits, well, all bets are currently off.**
5. **Last month, I said I’d have more details about LrcSets and LRCs, in particular MBeta. MBeta is tricky because there are two “formulae.” To simplify the discussion, assume that we have a left/right sensor so that there is a “left formula” and a “right formula.” The discussion for Up/Dn sensors is similar. With two formulae, I must know not only when the particle passes the filter, but then also which formula to apply; for inverseCube and logit, there’s only one formula. This means that, in general now, the single function postFilterCpaToNmi needed to be replaced by 4. Jim has done preliminary testing and MBeta seems to work, although I doubt that he has tried asymmetric MBetas with non-zero different minRanges and different pod values.**
6. **I had to revamp my integration routine for finding the sweepWidth for an LrcSet, but that seems to be working better now.**
7. **Long discussions with Jack about projections, and how boxes are crafted in the face of bad numbers. Basically, I make a bunch of executive decisions. One of these caused Jim to assume that I’d made an error. I guess I had. The situation was this; you have a high minimum track spacing and a lot of track that you have to use and a small box. Either you blow the box out, risking enormous overlap problems or you cut down on the track or you cut down on the track spacing. My executive decision was the wrong one. I changed it.**
8. **Discussions with Melody and Rob about regression testing for Sim. Followed that with discussions with Jack and then Jim. The latter two agreed that exact replications of the particle file are not really necessary, so I wrote code that compares two particle files from the same case, and puts the comparisons into a spreadsheet. There is a line for each scenario/objectType/timestep combination. I presented this to Jim, Melody, and Rob, and I think it will be the basis of Sim testing. I also wrote a program that will take pairs of particle files and build a single spreadsheet, with one sheet for all of the pairs, and a separate sheet for each pair. By writing a spreadsheet, additional functions can be added.** **We’ll see how this goes.**
9. **I had some problems with the new code when I was given sw=0. This is an arcane concept that was used to indicate something or other, but I don’t recall what it was, and it was causing Young’s cases to crash. I got that fixed. I’d like to get rid of sw=0 entirely. We have LRC subtypes of Logit, InverseCube, and MBeta. Why not add a new one “Blind?” That would simplify my code.**
10. **In a similar vein, I had to revamp some code to continue to accommodate the “Christmas Tree” style of giving me sensors; I don’t think that anyone is yet ready to give me an NvgEss sensor as two sensors and, until they are, I have to keep the old style to make sure that “everything that used to work still does (including NvgEss).”**
11. **Cordell found a crash in 2.1.1. I got the case and noticed that the cause was that I was being given an object type that had no lateral range curve. Now, by using LrcSets instead of a LateralRangeCurve, I notice that the set is empty and PFail starts at 1 and stays there as I process the (empty) set of LRCs. In the old case, we would try to find PFail from a null LRC. So no, it’s not good that there is an object types with no LRC(s), but 2.2 handles it.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Activity Worked** | **Hours Worked** | **Hourly Cost** | **Total Cost** |
| Kratzke | Coding/Doc/Travel | 144.19 | -- | -- |
|  |  |  |  |  |
| **Totals** |  | 144.19 |  |  |
|  |  |  |  |  |