

12/4/08 TIM Supplemental Questions:

1. Procedures and Bundles: (From Ashley Kells)

- a. What is the difference between a CAVE Procedure and Bundle?
- b. Why are bundles much larger than Procedures in size?
- c. What is the difference between the Procedures and Bundles that you can save under the CAVE menu vs the File menu when in the D2D Perspective?

2. Application Migration:

- a. To our understanding, LAPS, MSAS and LDAD will not be ported, but wrapped in AWIPS II. How will things be changes since these software pieces read and write netCDF files??? (Ashley Kells)

3. Camel Question

- a. Why did Raytheon choose Camel and not Servicemix as the new ESB? Camel is really not a true ESB (just a messaging service). Servicemix is produced by apache as well and is considered a true enterprise service bus. **(from Jim Williams)**
- b. How much granularity is anticipated for the log files which log the data ingest and data transactions? In the Mule era, it seems that the approach is to log information into "mega", monolithic type files instead of having log files for different decoders, for example. Maybe this is changing in the Camel era, but the ability to easily diagnose and monitor data handling is critical and it seems that more-tailored log files would help in this regard. **(Mark Glaudemans)**
- c. Does Raytheon plan to assign any resources to the Apache Camel project? In trying to implement a small application, I found the JavaDoc for Camel to be missing or to be lacking detail, and the Camel documentation was not complete enough to successfully implement message routing for a non-trivial configuration. So it appears that the Camel project is not very mature at this time, and it may require expert knowledge to support implementing the services layer of AWIPS-II. **(From Thomas Kretz)**
- d. What training on Camel will the NWS receive? **(From Thomas Kretz)**
- e. Can Raytheon provide performance test data comparing Camel and Mule? **(From Thomas Kretz)**
- f. Why did Raytheon choose to use the Spring XML route configuration instead of the Java DSL configuration as recommended by the Camel project? **(From Thomas Kretz)**

4. Hydro Questions (From Mark Glaudemans)

- a. For the hydro MPE precipitation processing, will the "xmrg" (a homegrown binary) format still be used to store the generated precipitation grids? It is not clear to what extent the MPE (multi-sensor precipitation processor) will be re-engineered. Earlier statements from Raytheon indicated that the MPE_FieldGenerator application will be left as is, which leads us to think that the generated files will remain in xmrg format. **(From Mark Glaudemans)**
- b. Are there published details on how external applications can read and write HDF5 format files using either Java, C, or other languages? **(From Mark Glaudemans)**
- c. "Since standard exporting of displays of Java applications can be painfully slow, what kind of infrastructure will Raytheon be putting in place to make sure that the user interfaces of Java applications can be run remotely during support activities?" **(From Mark Glaudemans)**
- d. Also, in the Hydro section of the agenda, the word "generics" is used. That should probably be changed to something else, since in Java, "generics" is a term used to describe a feature that is similar to C++ templates. The word might just spark a confusing 5-minute discussion in the wrong direction. **(From Mark Glaudemans)**

5. CAVE question

- a. At the previous TIM Frank mentioned in passing that a "headless CAVE" was being developed. Can we find out more about what that will be and how it will work? In particular, would that provide a mechanism for scripting CAVE for automated testing? **(from Stowe Davison)**

6. General (Shared libraries)

- a. Have you considered a unified architecture with shared libraries for user extensions such as Smart Tools, Derived Parameters, Text Products, WarnGEN, GHG?
Would it be possible to have more substantive TIM discussions between the AWIPS I and AWIPS II developers? **(From Tracy Hansen)**

Explanation:

It appears that T09 introduces designs for the following features:

- Derived Parameters implemented with a specialized Python user interface and library of functions
 - WarnGEN implemented using Velocity templates
 - Smart Init implemented with a specialized Python user interface and library of functions (as in AWIPS I)
 - Smart Tools implemented with a specialized Python user interface and library of functions (as in AWIPS I)
 - Text Products implemented with a specialized Python user interface and library of functions (as in AWIPS I)
1. We have not heard yet about GFESuite Procedures or iTools (required for automated Text Product testing). Will they also be implemented with independent specialized Python interfaces as in AWIPS I?

In AWIPS I, all these features evolved independently as forecasters learned to operate in the new digital paradigm. Looking back, we now see that they all embody different aspects of data translation -- from grid to grid or grid to text, and as such, lend themselves to a unified interface and common library of functionality. We understand that the goal of AWIPS II is to replicate the current AWIPS I features and therefore needs to maintain the existing Python user interfaces. We are pleased to see this happening in AWIPS II so that forecasters will be able to seamlessly port their AWIPS I Smart Tools, Procedures, Smart Init and Text Products to AWIPS II.

However, since new architectures and designs are being built to implement features such as Derived Parameters and WarnGEN, it would be cost-effective to build them in a unified way with shared underlying libraries. For example, it should be possible to call "getGrids" from a Text Product or "makeRow" from a Smart Tool. This would then allow for upcoming parts of AWIPS I functionality (Hydro, NAWIPS) to be more easily implemented and make the system more flexible, maintainable, and trainable.

Have you considered this type of approach? Related question: The AWIPS I developers are deeply familiar with the existing features and functionality, and perhaps have insights into how the AWIPS II architecture could be effectively designed. Would it be possible to have more substantive discussions and/or reviews with them

7. General

- a. What will be missing from D2D in TO10? (from Woody Roberts)