OPEN NAVIGATION SURFACE WORKING GROUP

TELECONFERENCE SUMMARY

17 January 2006
Hosted by SAIC Teleconferences

DRAFT VERSION
January 17, 2006

1. Introduction

This document details the results of the teleconference held at 1000EST on 2006-01-17 to discuss the remaining details of the Open Navigation Surface Working Group's (ONSWG) effort towards a candidate release and subsequent V1.0 release of the ONS library (a.k.a. BAG library). See section 5 for a participant list. The teleconference was kindly hosted by Shannon Byrne via the SAIC teleconference facility.

In the following, names of people with action items are shown in **BOLD SMALL CAPS**; expected deadline release dates are shown in red. Sizes of variables are indicated by 'U' for unsigned, 'S' for signed, 'F' for floating-point, and a size in bits (e.g., U8 is an eight bit unsigned integer, F64 is a 64-bit (double precision float-point number). Data sizes are given in bytes (B) with the usual convention that the SI multipliers are taken to mean multiples of 2^{10}B (i.e., 1kB = 2^{10}B = 1024B).

2. Summary of Discussion

2.1. Tracking List

> *BACKGROUND: The tracking list is the component of the BAG that allows the hydrographer to make changes to the base grid (a.k.a. designated or golden soundings), record why they were made and what the original contents of the grid were. Recording the original contents makes the base grid fail-safe: it is always what the hydrographer considers to be 'safe', and you have to reverse the modifications in order to get to what was originally predicted by any algorithm.*

Primary issue is the contents of the tracking list; all agree that this needs to be in the candidate release (CR), so that the full base functionality of the BAG is available. After some discussion, the agreed core contents are:

- Row, column location of the node of the BAG that was modified (two U16s).
- The original (depth, uncertainty) that the node contained (two F64s).
- A reason code indicating why the modification was made (a U8).
- An index number indicating the item in the metadata that describes the modifications (a U16).

This results in a fundamental size of 23B. The intent of the index number element is to allow modifications to be grouped together and linked to a more general description of the hydrographer's action at the level of the file metadata. This allows the data lineage section of the metadata to refer to a cluster of modifications and provide some descriptive text, e.g., "Modified CUBE node results to agree with hydrographer's interpretation of least depth", etc.

After discussion, this was adopted by **MCDONALD** for action by the CR date (see section 2.9)

2.2. X/Y Arrays of Location

> *BACKGROUND: When data is read from the BAG, at present the user has to compute the offsets of the nodes in order to be able to compute with the data. This is error-prone.*

Following discussion with a sub-group of developers at the Shallow Survey 2005 conference in Plymouth, UK, it was suggested that the library should provide location computation services so that the user-level code (a) doesn't have to do this, and (b) doesn't do it wrong (e.g., confusing the difference between node-based and grid-based locations and therefore always being half a node off). The outline was for the library to provide two 1D arrays of locations, one indexed by row of the data returned, and one indexed by column; it is assumed that the user code can do the look-up by (row, col) successfully.

After some discussion, it was agreed that this was a workable solution, but that to avoid difficulties, the following conditions would be required:

- There will be two call sequences for getting data from the BAG:
  - Call for data only; returns simple 2D arrays of depth and uncertainty information only (arrays of F32s in row major order as it is in the file).

    - Call for data with locations; returns simple 2D arrays of depth and uncertainty as above, and two 1D arrays of locations for the (subset of) data returned using the coordinate system used internally in the BAG file itself.
- There will be no way to determine locations after the fact. This avoids having to hold state either in the library or through some user held structure.

The issue of restricting the locations to the coordinate system of the BAG itself was discussed, and it was agreed that this is not something that we would prefer to have as the final solution, simply a stop-gap to get to CR. The best solution would be to have the library do projections generally using the GEOTRANS interface, but it was thought that this would add too much complexity for us to put in before CR and should be deferred until afterwards.

After some discussion, this was adopted by **MCDONALD** for action by the CR date (see section 2.9)

2.3. API to Help in Troubleshooting

There is presently no way to decode error numbers that are returned from the BAG library, which makes debugging a little complex. This was an issue particularly when the `BAG_HOME` environment variable is not set, as was found by some integration attempts with development users.

There are a number of potential solutions for this, but the simplest was agreed to be:

- Provide a routine to translate error codes into text descriptions.
- Provide a check that the BAG_HOME variable is something reasonable when the BAG is first opened to avoid this falling apart at some random time further into the process.

After some discussion, this was adopted by **MCDONALD** for action by the CR date (see section 2.9)

2.4. Metadata Update for Vertical Uncertainty

> *BACKGROUND: The BAG supports a general 'uncertainty' layer intended to represent the uncertainty of the elevations in the data. However, there is no particular representation required in the uncertainty, since this can be different in different BAGs, and may change in the course of the lifetime of a BAG (e.g., when it goes from source data to product surface). The representation should be in the metadata.*

The question is primarily how to represent the description of the uncertainty in the metadata: a general text description (very general but subject to abuse) or an enumerated type (simple, but inflexible). Discussion (Depner) suggested that the latter is a more reasonable solution because it can be handled algorithmically, and if we arrange to have the system flexible enough we can add shades of description if required. It was therefore agreed to have the uncertainty described by an integer (U32) in the metadata, and have the verbiage linking this number to a human interpretation in the documentation of the BAG. As a starting point, the following should be defined:

0. Unknown.
1. Raw standard deviation of soundings the contributed to the node.
2. Standard deviation of soundings captured by a CUBE hypothesis (i.e., CUBE's standard output of uncertainty).
3. NOAA standard product uncertainty V1.0 (a blend of CUBE uncertainty and other measures).
4. Estimated standard deviation based on historical/archive data.

Other values can be defined as required using the same mechanisms as required for modifications to the BAG specification (as defined in the FSD).

After some discussion, this was adopted by **LAMEY** for action by the CR date (see section 2.9)

2.5. GEOTRANS Requirement for Candidate/1.0 Release

> *BACKGROUND: The GeoTrans library provides general coordinate transform services and has been integrated into the BAG library to assist users in translation between projected BAG data and geographic coordinates.*

After discussion, it was agreed that there was no need to do a direct integration of the GeoTrans code into the core library routines, although this was essential for future development and should be prioritized as soon as possible after the release process. However, in order to help with user acceptance, it was agreed that there should be some example code provided with the library release that will allow this to happen (this is a bigger issue and is covered below in section 2.10).

2.6. OS Support for Build and Testing

> *BACKGROUND: The BAG FSD recommends that we support at least Win32 and Linux platforms, and as many other UNIX style systems as we can.*

The development group already supports Win32 (XP) through CARIS involvement (**LAMEY**) and Red Hat Enterprise through SAIC involvement (**BYRNE/ MCDONALD**). It was agreed that this was sufficient to test the system without further development of testing protocols for now. Depner agreed to find out if he could put together an Open Solaris configuration to test against (some at NAVO are still using this for work, and want to use BAG). (Action: **DEPNER**). Riley agreed to test on Win32/Windows 2000, and a Win64 platform that he has to check for 64-bit clean pointers, etc. (Action: **RILEY**.) Finally, Byrne agreed to check with Moggert at 7Cs since they support a number of other platforms, and might be able to contribute some other tests. (Action: **BYRNE**, **MOGGERT**) (Moggert was not able to take part in the teleconference due to technical difficulties.)

2.7. Configuration Management

CM had been a problem during earlier development due to access restrictions at many of the vendor sites where firewalls prevented access to the CVS server at CCOM/JHC. CARIS and IVS confirm that they can reach the CVS server now, and SAIC appear to have a network connection that they can use, but need to test (Action: **BYRNE**). NAVO cannot do CVS and have confirmed that they will never be able to do this because of network restrictions on their new internet configuration. They can do HTTP however, and this might be a workaround – need to identify a software suite that would allow this (Action: **CALDER**). There are other workarounds, e.g., access from home. This is readily supported with the CVS server as it stands now, only needs the IP addresses of the machines needing access sent to Calder (Action: **DEPNER**, **LADNER**, **FABRE**).

One concern about generic CVS is that the mail notification might be a little too coarse (e.g., when a release is tagged a message gets generated for every file). Riley recommended CVSmailer, which is a little more nuanced and should be investigated (Action: **CALDER**).

2.8. Release Mechanics

> *BACKGROUND: There is a web site, http://www.opennavsurf.org, hosted for the project by CCOM/JHC.*

The web-site can be used for delivery of the distribution for both the candidate and general releases of the project code. It was agreed that the person downloading the distribution (as a tarball) should provide some information, limited to a contact e-mail and a comment on their intended use of the source. This will need to be added to the web-site (Action: **CALDER**). There was some discussion about whether we should release binary builds of the project or not. Most felt that this would be beneficial although the primary aim is still to ensure that the library will build from the raw source directly. It was agreed that the binaries that the development group build themselves should be contributed to the CVS tree under a separate project, and distributed from there via the website; however, to keep things simple, the CR will have only source release (Action: **BYRNE/MCDONALD**, **LAMEY**, **DEPNER**, **MOGGERT**, **RILEY**, **CALDER**).

2.9. Release Dates

After looking at the work outline, it was agreed that the CR could be managed by 2006-02-03 (a Friday), and that anyone interested could provide comments on this until 2006-03-03 (a Friday). The first full release (FR) will occur on 2006-03-31 (a Friday). This gives four weeks for comments, and four weeks to incorporate the comments before the FR, a reasonable arrangement.

2.10. Example Programs

> *BACKGROUND: A request has been received from LCDR Shep Smith, NOAA, that the library should have at least an example program that allows a BAG to be read and*

translated into something that standard GIS tools can read. This will ensure that the early-adopter users will be able to use BAGs in their standard tools before they get a full BAG implementation. This will also be useful into the future for 'other use' aspects of the BAG's goals.

It was agreed that this was a good idea, and would also satisfy the requirement for getting geographic information out of a BAG that is projected if done correctly. It was agreed that the code would read a BAG and dump the contents to:

1. Plain ASCII in the native coordinate system of the BAG.
2. Plain ASCII in geographic coordinates if the BAG is projected.
3. ArcView compatible ASCII in native coordinate system of the BAG.
4. ArcView compatible ASCII in geographic coordinates if the BAG is projected.

It was agreed that projecting the BAG into an arbitrary projection system was more than this tool should achieve in the timescale to CR, and in any case we want that to be part of the library in the end and this would be counterproductive. After some discussion, the development of this tool was adopted by **FABRE** for action by the CR date (see section 2.9).

It was agreed that we will also need an example BAG to distribute with the library, but it was felt that the example of the Shallow Survey 2005 dataset might be a little too big. It was agreed that we could use a subset of this, and that this should wait until the final version of the CR library was built.

2.11. Other Business

The only other business was a question from McDonald on the metadata for location of the BAG. It appeared that this information was only good to 2 d.p. in geographic coordinates, which is inadequate for almost all uses. Lamey confirmed that this section of the metadata is only really intended to provide a general location of the area under investigation, and not fine-scale detail is preserved elsewhere. The idea here is that the general information gives an overall rectangle, while the fine-scale detail could be something more general (e.g., a polygon). The fine-detail information has 6 d.p. (~0.11m), which should be adequate for most uses but might need to be extended (to, e.g., 8 d.p., ~0.11cm) for very high resolution grids.

3. Summary of Action Items and Dates

The following actions and dates were agreed:

Person	Action(s)	Section	Date
Byrne	Build and test Red Hat Linux binaries	2.6	CR
	Check with Moggert on other UNIX test platforms	2.6	ASAP
	Test CVS download	2.7	ASAP
	Organize next teleconference and send details via e-mail	4	2006-01-27
Calder	Check on availability of HTTP access to CVS server	2.7	ASAP
	Check on availability of CVSmailer	2.7	ASAP
	Make binary release CVS project	2.8	FR
Depner	Build and test Open Solaris binaries	2.6	CR
	Provide IP address of machine for CVS access to Calder	2.7	ASAP
Fabre	Provide IP address of machine for CVS access to Calder	2.7	ASAP
	Example program for BAG to ASCII/ArcView with unprojection	2.10	CR
Ladner	Provide IP address of machine for CVS access to Calder	2.7	ASAP
Lamey	Implementation of vertical uncertainty meta-data enumeration	2.4	CR
	Build and test Win32 (XP) binaries	2.6	CR
McDonald	Implementation of the tracking list	2.1	CR
	Implementation of the X/Y location array API	2.2	CR
	Implementation of the error reporting API	2.3	CR
	Build and test Red Hat Linux binaries	2.6	CR
Moggert	Check with Byrne on other UNIX test platforms	2.6	ASAP

Person	Action(s)	Section	Date
Riley	Build and test on Windows 2000	2.6	CR
	Build and test with Win64 binaries	2.6	FR

The release dates were agreed as:

- Candidate: 2006-02-03
- Comments: 2006-03-03
- Full: 2006-03-31

4. Next Meeting

It was agreed that there should be a further teleconference before the CR date, and that the best date was 2006-01-27 at 0800PST, 1100EST. **BYRNE** to arrange and communication details to the development group via the e-mail list.

5. Participants

Brian Calder (CCOM/JHC)
Bill Lamey (CARIS Ltd)
Shannon Byrne (SAIC Newport)
Webb McDonald (SAIC Newport)
Dave Fabre (NAVOCEANO)
Wade Ladner (NAVOCEANO)
Jan Depner (NAVOCEANO)
Jack Riley (NOAA HSTP)