TSMAD26/DIPWG5-08.1G

US Navy SPAWAR ATLANTIC TSMAD/DIPWG Comments

PresLib_e4_Part_I _draft_v0.2_Master.doc

General Comments

- Flow charts/sequence diagrams for the conditional symbology procedures are an improvement over the Nassi–Shneiderman diagrams of earlier versions.
- The C program for conditional symbology procedure LITDSN being provided is mentioned in several places in the document although TSMAD26_DIPWG5-8.1A_Changes_to_S-52_Presentation_Library_to_Produce_v4.0.doc says C code is no longer being provided.

Section 4.3

- There needs to be clarification on what is meant by “a limited digital version in .dai format is provided…”
- It would be helpful if more definitive information on the future availability of the .dai files was provided. We would recommend removing it completely and replacing it with XML files. XML files would be easier to deal with than parsing the .dai file or manually extracting the tables from the word processing version.
- The inclusion of C code is still mentioned for LITDSN.

Section 5

- Recommend replacing Figure 1 with the same style of flowchart diagrams used by the CSPs.

Section 8

- Comment [p5] is correct, the AA instruction is not used at all by the Presentation Library.
- The PM instruction could benefit from a diagram.

Section 8.4.2

- Not sure the assumption that most ECDIS systems will use the stipple fill technique is warranted, therefore the justification for 25%, 50%, and 75% fills are invalid. Alternatively, the stipple fill patterns could be based on ranges of transparency percentages.

Section 8.5.1

- Since the center of gravity algorithm given in PL 3.4 has been removed from the spec, a consistent algorithm should be developed for centered symbols instead of burdening the ECDIS manufacturers to implement their own. In addition to handling the case where the center of gravity is not on the area, the algorithm also needs to handle the case where the computed point is very close to the edge of the area. This is needed to prevent confusion as to which adjacent areas the symbol belongs to. This algorithm should also ensure that the calculated point remains stable as possible as the area changes its shape as it enters and leaves the screen.

- Guidance needs to be provided in the case where a single area shows up as two or more separate areas on the screen due to truncation. That is, should the symbol be displayed on each of the areas or just one? If just one, which one (the largest, the one closest to middle, some other criteria)?

Section 8.5.4+

- Numbering after this section is incorrect.

Section 13.1.6

- Should units of measure be a mariner selection? If so, how should measurement based symbology (such as soundings, certain text, or spatial queries) be expanded to handle this. Care also needs to be taken to ensure any conversions err on the side of safety. For instance, depth measurements need to be rounded down to ensure they don't indicate a deeper measurement than what the data indicates.

Section 13.2.21

- How would English units (e.g. feet) be symbolized?

S-100 / S-101 Portrayal in General

Portrayal Catalog

According to TSMAD26_DIPWG5-10.1A_S-101_RiskRegister-April2013.pdf, it appears that the portrayal catalog is behind schedule. The hold-up seems to be the effort involved in expressing the CSPs in a form that is compatible with XSLT. Although XSLT is Turing-complete, it is a declarative language as opposed to an imperative language. Since the existing CSPs are expressed in imperative form, the difficulty encountered in expressing them in declarative form is understandable.

There is mention of hardwiring functions needed for the CSPs into the XSLT parser to overcome this difficulty. All this will accomplish is moving difficult portions outside of XSLT and into a presumably imperative language that the XSLT parser itself is written in.

It might be worthwhile to reexamine the whole XML / XSLT portrayal model and perhaps create something simpler; such an enhanced S-52 .dai file expressed in XML. This would allow a richer table lookup and would allow the SVG commands to exist side by side with the lookups.

As an example, the RESCSPnn CSP could be replaced by extending the attribute value lookups to allow matching out of a list and nesting. More complicated symbology can still be accomplished using CSPs in a manner similar to S-52. A notional XML file showing these examples is given at the end of this document.

-

Symbology XML Example

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- Notional example of complex table lookup -->
<library>
  <symbols>
    <symbol name="ENTRES51">
      <!-- SVG Commands here -->
    </symbol>
    <symbol name="ENTRES61">
      <!-- SVG Commands here -->
    </symbol>
    <symbol name="ENTRES71">
      <!-- SVG Commands here -->
    </symbol>
    <symbol name="INFARE51">
      <!-- SVG Commands here -->
    </symbol>
    <symbol name="RSRDEP51">
      <!-- SVG Commands here -->
    </symbol>
  </symbols>
  <lookups>
    <lookup feature="FAIRWAY">
      <attribute_present attribute="RESTRN">
        <call procedure="RESCSP02"/>
      </attribute_present>
    </lookup>
    <lookup feature="WRECKS">
      <attributes_present attribute="DRVAL1,WATLEV">
        <call procedure="UDWHAZ04"/>
      </attributes_present>
    </lookup>
  </lookups>
  <procedures>
    <procedure name="RESCSP02">
      <match_any attribute="RESTRN" values="7,8,14">
        <!-- Continuation A in CSP -->
        <match_any attribute="RESTRN" values="1,2,3,4,5,6,13,16,17,23,25,24,26,27">
          <symbology command="SY(ENTRES61)"/>
        </match_any>
        <match_any attribute="RESTRN" values="9,10,11,12,15,18,19,20,21,22">
          <symbology command="SY(ENTRES71)"/>
        </match_any>
        <match_default>
          <symbology command="SY(ENTRES51)"/>
        </match_default>
      </match_any>
      <match_any attribute="RESTRN" values="1,2">
        <!-- Continuation B in CSP -->
        <match_any attribute="RESTRN" values="3,4,5,6,13,16,17,23,24,25,26,27">
          <symbology command="SY(ENTRES61)"/>
        </match_any>
        <match_any attribute="RESTRN" values="9,10,11,12,15,18,19,20,21,22">
          <symbology command="SY(ENTRES71)"/>
        </match_any>
        <match_default>
          <symbology command="SY(ENTRES51)"/>
        </match_default>
      </match_any>
      <match_any attribute="RESTRN" values="3,4,5,6,24">
        <!-- Continuation C in CSP -->
        <match_any attribute="RESTRN" values="13,16,17,23,25,26,27">
          <symbology command="SY(ENTRES61)"/>
        </match_any>
        <match_any attribute="RESTRN" values="9,10,11,12,15,18,19,20,21,22">
          <symbology command="SY(ENTRES71)"/>
        </match_any>
        <match_default>
          <symbology command="SY(ENTRES51)"/>
        </match_default>
      </match_any>
      <match_any attribute="RESTRN" values="13,16,17,23,25,26,27">
        <!-- Continuation D in CSP -->
        <match_any attribute="RESTRN" values="9,10,11,12,15,18,19,20,21,22">
          <symbology command="SY(ENTRES71)"/>
        </match_any>
        <match_default>
          <symbology command="SY(ENTRES51)"/>
        </match_default>
      </match_any>
      <match_any attribute="RESTRN" values="9,10,11,12,15,18,19,20,21,22">
        <symbology command="SY(INFARE51)"/>
      </match_any>
      <match_default>
        <symbology command="SY(RSRDEP51)"/>
      </match_default>
    </procedure>
    <procedure name="UDWHAZ04">
      <symbology command="CS(UDWHAZ04)"/>
    </procedure>
  </procedures>
</library>
```