Synthesis of Findings and Recommendations from Evaluation of DIGGS 1.0a Schema

October 6, 2009

Contact:

Loren L. Turner, P.E.
Senior Transportation Engineer
Caltrans, Division of Research & Innovation
5900 Folsom Blvd. MS-5, Sacramento, CA 95819
(916) 227-7174
loren.turner@dot.ca.gov

Prepared For:

Transportation Pooled Fund Project TPF-5(111)
"Development of Standards for Geotechnical Management Systems"

1 Introduction

In the Summer of 2009, the DIGGS Project Team contracted with Galdos Systems and Compusult Ltd. to carry out independent reviews of DIGGS version 1.0a schemas. The objectives of this work were to:

- Assess the current implementation of DIGGS with GML. (Were GML standards, conventions, best practices, and patterns implemented correctly? Was the appropriate version and/or variation of GML used based upon the requirements of DIGGS?)

- Assess if the goals of the DIGGS standard is best implemented as a GML application schema, or if other encoding standards (both GML and non-GML) should be considered.

- Assess the strategy that was used in the organization and composition of schema objects and files in the context of the physical observations and data that DIGGS is trying to capture. (Is the schema too decomposed, or appropriate given the requirements of DIGGS?)

- Assess the core issues recently documented by the DIGGS Project Team on the discussion forums and at the March 2009 DIGGS meeting in Orlando.

- Consider other issues, as appropriate, that may not have identified by the DIGGS Project Team, that exist with the schema design that may impact its application.

Final reports from Galdos and Compusult with findings and recommendations were delivered to the DIGGS Project Team. Both reports can be found on the DIGGS website (http://www.diggsml.com/diggs-project-team-contracts-galdos-systems-and-compusult-ltd-carry-out-review-diggs-version-1-0a). This document synthesizes the recommendations from the two reports and presents a draft scope of work for the next phase of work.

2 Galdos and Compusult Findings

The reports from Galdos and Compusult provided a comprehensive review of the DIGGS v1.0a schema and identified a number of significant issues that need to be resolved. Although the reports had different areas of emphasis, and, in some cases, differing recommendations on particular issues, both reports concluded that the DIGGS schema was not ready for deployment.

- *"It is our conclusion that the DIGGS 1.0a schemas are currently not ready to be deployed in anything other than an experimental environment."* (Burggraf, D. 2009. "DIGGS 1.0a Schema Evaluation." Report from Galdos Systems to the University of Florida, March 13, 2009)

- *"We suggest that the DIGGS schemas are not yet ready for official release."* (Mitten, P. 2009. "DIGGS 1.0a Schema Evaluation." Report from Compusult Limited to the University of Florida, August 6, 2009)

Combined, the reports identified 11 key issues and provided recommendations on how to fix the issues, the relative priorities of the issues, and an estimate of the resources required to carry out the work. The issues are as follows:

- GML "Object-Property" Patterning
- Inheritance of Objects
- Use of GML Profiles
- Organization of DIGGS Schemas
- Relative Referencing of Schemas
- Update to GML 3.2
- Code Tables
- Key Fields
- Table Data
- Validation Using Web Services
- Schema Management Tools

Several issues and recommendations were not included in the list above, since the list reflects a fundamental recommendation by the Core SIG to continue the development of DIGGS as a *GML Application Schema*. The Compusult report offers a number of constructive recommendations to remove many of the GML-specific requirements which, in turn, could potentially simplify the DIGGS schema. However, those recommendations would also render DIGGS unusable by general purpose applications that currently support GML standards.

In a telephone meeting of the Core SIG on September 24, 2009, the merits of continued GML compliance for DIGGS were carefully considered, and the team concurred that the benefits of GML outweighed the downsides at this time. Furthermore, the Core SIG felt that the perceived complexity associated with GML in DIGGS v1.0a was not entirely accurate due to a number of errors in how GML was implemented in that version.

3 Identified Issues

Detailed descriptions of identified issues are presented in the table below. Issues were assigned metrics to convey their significance and priority relative to the other issues listed. Since Galdos and Compusult each used different metrics and had a different approach to prioritization, generating this combined list was the result of grouping similar recommendations and prioritizing the issues to the extent possible.

Task	Description	Estimated Resources
1	**GML "Object-Property" Patterning** Fix errors due to several design patterns that do not conform to the GML 3.1 Object-property rules. Existing pattern errors include property types that extend a GML object type, property elements typed as a GML object type, property types with non-homogeneous content, elements that substitute for "gml:_Object," and geometry type declaration. *(Reference – Galdos Final Report, Section 5.1.2, "GML Conformance")* *(Reference – Compusult Final Report, Section 4.1, "Were GML Standards, Conventions, and Best Practices Implemented Correctly?")*	8 days
2	**Update to GML 3.2** Migrate the DIGGS GML 3.1.1 application schema to a GML 3.2 application schema. *(Reference – Galdos Final Report, Section 5.4, "GML Version")*	2 days
3	**Inheritance of Objects** The DIGGS schemas should be refactored to employ inheritance only where it is used polymorphically. Inheritance should be used only when defining specializations of a general concept where it would be reasonable to use the general object as a placeholder. However, it should not be used as a shortcut for adding properties of one object to another, especially if not all of the properties of the parent object are appropriate. *(Reference – Compusult Final Report, Section 2.11, "Inheritance of Objects")*	5+ days
4	**Use of GML Profiles**	2 days

	Use a GML profile to lower the number of elements and types to import from GML, and restrict undesirable attribute and property occurrences that are optional on GML types. *(Reference – Galdos Final Report, Section 6.1.1, "GML Profile")*	
5	**Organization of DIGGS Schemas** Repackage the DIGGS schemas to one file per namespace to reduce the excessive number of XML parser import/include calls. *(Reference – Galdos Final Report, Section 5.2, "Schema Packaging and XML Processing Impact")*	5 days
6	**Relative Referencing of Schemas** Eliminate the OASIS XML Catalog requirement by: (1) updating the DIGGS schemas to use relative schemaLocation references, as is used in the GML and WITSML schemas; (2) updating the DIGGS examples to not specify a schemaLocation attribute; and (3) including supporting documentation that instructs users how to use their XML IDE to validate the examples. *(Reference – Compusult Final Report, Section 2.3, "Recursion of DIGGS Objects")*	1 day
7	**Code Tables** Adopt an extendable XML Schema enumeration pattern for code tables. This approach defines the "standard" values for a code table in the schema and provides an optional extension mechanism. Translate the code tables already used by DIGGS and update the schemas such that the lists are specified appropriately. *(Reference – Compusult Final Report, Section 2.5, "Code Table Implementation")*	3 days
8	**Key Fields** Update the DIGGS schemas to use the <key> element. *(Reference – Compusult Final Report, Section 2.6, "Key Fields / Use of Unique Identifiers")*	3+ days
9	**Table Data** Implement a general structure for table data that explicitly defines each "standard" column expected in the table and, if necessary, allow a repeatable generic column in the structure. *(Reference – Compusult Final Report, Section 2.4, "Table Object Implementation / Table Object Data Validation")*	2+ days
10	**Validation Using Web Services** Employ validation that goes beyond schema, since XML schema alone cannot enforce all validation rules for DIGGS. Deploy a web service that would accept a	30 days

	DIGGS data instance and return a report of errors and warnings that is easy to read and understand. *(Reference – Galdos Final Report, Section 6, "DIGGS Schema Roadmap")*	
11	**Schema Management Tools** Schema version management, governance procedures and the assignment of roles to maintain the life cycle of the DIGGS schemas should be formally documented and disseminated to the DIGGS committee. A Revision Control System (RCS) should be implemented for DIGGS Schema development to provide a history of revisions, with comments regarding changes, concurrency control, and the ability to diff revisions and merge files in a local repository via distributed updates. *(Reference – Galdos Final Report, Section 5.5.1, "Schema Registries/Repositories")* *(Reference – Compusult Final Report, Section 5.0, "Web Based Community Schema Development")*	15 days

4 Future Task Planning

The resource estimates provided by Galdos and Compusult to work through all 11 issues adds up to a minimum required 76 person-days. This estimate does not include any time or resources required by the DIGGS Core SIG or other external participants that may be needed to develop domain-dependent deliverables. (For example, in order for work to proceed on the "code table" issue, the DIGGS team would need to first develop and provide the enumerated lists and where they are to be used throughout the schemas.)

One possible approach for the next series of tasks is presented here. The work would be staged in multiple phases, dividing the work into three categories: (1) domain-independent work; (2) domain-dependent work; (3) and schema tools.

4.1 Domain-Independent Schema Work

This work would focus on sub-tasks that have little or no domain dependencies and would not likely require substantial work on the part of the DIGGS Core SIG prior to implementation. It is anticipated that these sub-tasks could be carried out by a qualified contractor with GML expertise with some guidance from the DIGGS Project Manager. These sub-tasks (as described earlier in this document) include:

- GML "Object-Property" Patterning
- Update to GML 3.2
- Inheritance of Objects
- Use of GML Profiles
- Organization of DIGGS Schemas
- Relative Referencing of Schemas

4.2 Domain-Dependent Schema Work

There are three issues in particular that will likely require significant involvement by the DIGGS Core SIG prior to implementation into the schema:

- Code Tables
- Key Fields
- Table Data

In-person or web-based meetings, facilitated by the schema development contractor, over multiple days would likely be needed to establish a standardized approach to the

implementation of these features, and develop the default code lists and table “column headers” for the various DIGGS objects.

4.3 Schema Tools

Two of the proposed sub-tasks focus on the development of tools and mechanisms to better manage and review the DIGGS schema:

- Validation Using Web Services
- Schema Management Tools

These sub-tasks would be carried out after completing the other schema work and would provide tools for the DIGGS Core SIG to conduct a cursory review of the schemas and provide a collective repository for DIGGS schema development.