D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

OPENGIS PROGRESS TOWARD INTEROPERABILITY IN PHOTOGRAMMETRY

Cliff Kottman and Louis Hecht
Open GIS Consortium, Inc.
ckottman@opengis.org
lhecht@opengis.org

ABSTRACT

The achievement of interoperability in photogrammetry is a difficult task. Today's photogrammetric services and servers tend to be tightly linked to proprietary environments, supported by heterogeneous platforms, distributed across networks not designed for shared image-related processes, and often are specific to one family of sensors. Interoperability, on the other hand, demands immediate access to any imagery, the image's geo-positioning information, and all photogrammetric processes that might display or otherwise exploit the image, whether these are distributed or heterogeneous or both, for the accomplishment of complex work flows in a single session.

The key to progress toward interoperability in photogrammetry is consensus in the marketplace at the level of software interfaces. There are two consequences of consensus interfaces: 1) interoperability (the ability of processing components to cooperate, even when they are designed independently of each other), and 2) commodity software components that will change the face of the marketplace. The interfaces must be carefully placed, however, if the photogrammetric software components are to thrive in the marketplace.

This paper reports on the progress made at the Open GIS Consortium, Inc. (OGC) toward market consensus on those interfaces. The organization and status of the Open GIS Consortium is introduced, and a vision of the photogrammetry and remote sensing marketplaces of the future is presented. The technology development methods of the Consortium are summarized, and a brief status report is given, with emphasis on the objects and methods already held in the consensus of OGC, and their relationships to photogrammetric objects. Examples showing how the Consortium is empowering its members to both guide and reflect the evolving digital image and remote sensing marketplace are given, with special attention to new infrastructure and new client services that open fresh market niches for entrepreneurs and that will enormously expand the size of the overall marketplace.

The remainder of the paper charts likely future directions of the Consortium. Highlighted are the candidate objects of photogrammetry: images and other grid coverages, stored functions, evaluators, mono- and stereoscopic photogrammetric models, accuracy and quality models, attributes and their values, metadata, photogrammetric processes, and their attributes and associations. Special attention is given to the methods (or "behaviors") supported by each class that enable interoperability. The OGC approach exploits the fact that images and other photogrammetric objects are subclasses of "GIS Feature," and so inherit many useful properties and behaviors from this and other more abstract object classes.

The paper presents two alternative approaches to the "stored function" objects that enable the communication of photogrammetric equations between clients and servers in a distributed environment, even between software components that were developed without knowledge of each other. The concept of class "folders" is also discussed, in terms of future "shrink-wrap software components" that can carry the OGC certification mark.

Finally, because interoperability is heavily dependent upon object-oriented design tools, we provide a summary of the software engineering tools in use at the Consortium, the role played by them, and the future impact of better object tools.

1 INTRODUCTION

Earth images, today, are not integrated into the global business culture, nor into any large marketplace. Earth images are isolated; used on magazine covers; looked at one at a time on an infrequent basis. Earth images are time consuming to download to your PC, and seldom worth the trouble. On your computer, they occupy a window of their own, seldom interacting with other images, and almost never interacting with other data types. Here, we are speaking mostly of literal earth images: photographs and digital images that are taken from airplanes or satellites. But this commentary on the limitations of earth images is just as valid for "processed" earth images: scanned maps, false color images, images formed of classed pixels, orthophotos, synthetic images, and so on.

The two main reasons that images cannot be used more effectively are 1) lack of technology availability and 2) lack of interoperability, which effectively spring from the same problem. With regard to earth images, interoperability means not just reconciliation of diverse formats, and not just a user's ability to quickly find and access useful images, but also network access to technology for viewing, registering, seeing one over another, managing footprints, supporting exploitation, etc.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

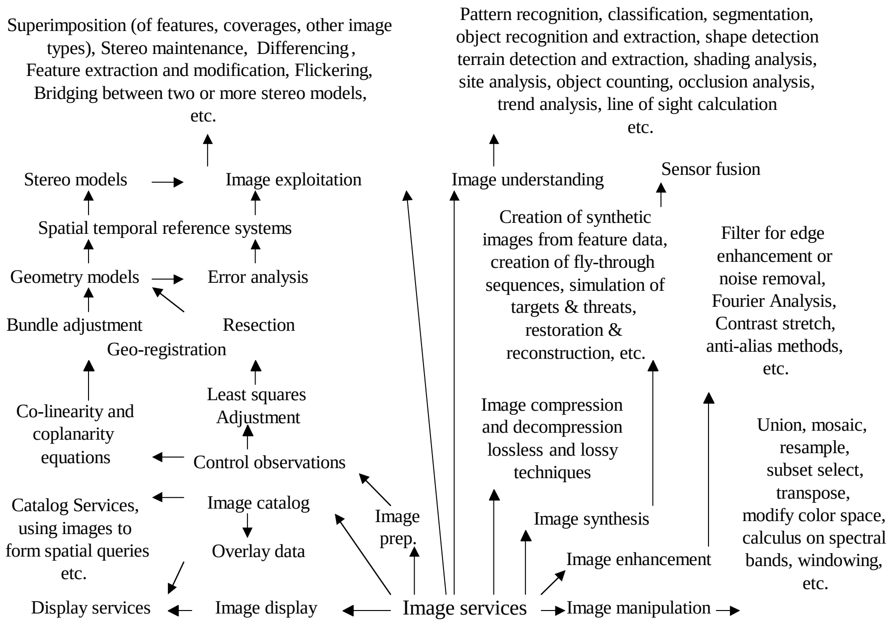

Figure 1. A Hierarchy of Image Services

In some isolated instances, such as in full-featured monolithic systems, and for narrowly defined data sets, many of these capabilities are available to a user. But if we accept the challenge to make earth imaging a part of today's highly distributed, Internet-based information environment and computing environment, it is clear that we must have interfaces that enable all kinds of systems to communicate in terms of images and image manipulation processes.

There is widespread confidence in the importance of earth imagery. In the US, Vice President Al Gore is using satellite imagery to advance his vision of an earth in more perfect balance with nature. Major corporations have been making large investments in earth imaging satellites and data delivery systems. We are witnessing the advent of real-time stereoscopy, the superimposition of anything on anything. Video will become more and more a part of information environments that use earth images. Spatial information catalogs will provide an index to global information of all kinds.

The point of the following graphic is that there are lots of things to do to open up the earth imagery world. Each box in the diagram holds dozens of technologies, dozens of interfaces. We need to chart our path carefully so that there is sufficient market for each release that the wealth generated by it will sustain the development of the next release.

Interoperability demands immediate access to any imagery, the image's geo-positioning information, and all photogrammetric processes that might display or otherwise exploit the image, whether these are distributed or heterogeneous or both, for the accomplishment of complex work flows in a single session.

The first step toward interoperability is to understand and share a common vision of the future, where interoperability is in place. Perhaps the "use case" is the most powerful means of capturing this kind of vision. Below we present three uses cases, three examples of how information flow ought to happen to support a particular task:

1. **The farmer:** *a.* Overlays a map of his fields with last year's imagery of his crop, last year's yield profiles, and this year's soil sampling profile; assesses the need for fertilizer; *b.* Overlays this with insect infestation imagery for the past 5 years, and assesses the need for insecticide. *c.* Overlays this with wetlands data from the county or state, and modifies his insect and fertilizer plans to protect a creek. *d.* Overlays this with a soil moisture profile, to plan irrigation. Notice the various actors: the farmer, the state extension service (with insect data), the Environmental Protection Agency, perhaps the Fish and Wildlife Service, the Farmer's Cooperative (with the algorithms for fertilization and insecticide), the image library (could be a commercial or government source) Notice the generation of wealth by the provision of new goods and services to the farmer. And notice how this use case clarifies what interfaces are important to get the scenario started.

2. **The home buyer** (in the broker's office, or on the home buyer's personal desktop): *a.* Selects a neighborhood, and looks at an aerial view of it. Houses for sale appear in red. Houses for sale in the desired price range and with mandatory features turn green. *b.* A few candidate green homes are selected. *c.* Using elevation data, a "fly through" allows the buyer to assess the view from each back porch. *d.* Perhaps a virtual walk-through of the rooms of the house is available. *e.* Using county imagery from 5, 10, 15, and 20 years ago, the buyer assesses the age of the roof, the health of the trees, and the trends in the neighborhood. *f.* Using city, county and state data, the imagery is superimposed with underground pipe and wire information; the assigned

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

school districts and schools are located. The nearest hospital and fire station are located, the lot lines are located, and the easements and rights of way are visualized. *g.* Using chamber of commerce data or images, the nearest shops and supermarket and auto repair shops are located. *h.* Using Automobile Association data, a family of alternate routes to work are charted, with the conditions that make each optimal. *i.* Using infrared sensitive imagery from the department of energy, assesses the heat loss in winter, and the need for new insulation. *j.* Using measuring tools, the size of the lot and house are independently measured. *k.* The size of the parking area is assessed and compared to the number of cars of the buyer. *l.* Using census tract data, the demographics in the neighborhoods are visualized. Again, note where goods and services are generated. Note the actors in the use case. Note that a large number of information-providing and process-serving actors are involved, but they must be transparent to the user.

3. **The soldier:** *a.* The soldier is on a peacekeeping mission. He uses information from surveillance cameras in planes, drones, and satellites. He receives video day and night in a continuous stream from lightweight airborne platforms. *b.* He passes the imagery through feature detection (looking for airplanes, rockets, artillery, etc.), and can automatically bring up yesterday's image whenever something is suspicious in order to reduce the number of false target features. If a spot wasn't covered yesterday, older imagery is provided and used. *c.* Images are registered to maps, so roads can be automatically monitored *d.* Image coordinates (pixel row and column) are automatically converted to the position data needed by his ordinance (that is, point and aim information). All the types of ordinance understand the target position objects; conversion is not necessary. *e.* The soldier is in contact with his command authority; they share a common view of the battle space. *f.* The soldier finds something of interest on an image. He points at it and generates a report instantly that is understood by other analysts, who check his finding using other sources. *g.* The other analysts are coalition members who speak a different language, and are supported with different technology. *h.* The soldier at other times makes contingency mission plans: selects potential targets, finds optimum access and egress paths, prepares flight folders that pilots and other weapon control officers can use to train, navigate, execute, escape, etc.

The point of such use cases is to: 1. Identify a wide range of actors (people and systems) and services (rectification, orthorectification, registration, differencing, feature detection, library services, image-to-image comparison, synthetic photo generation, etc.) 2. Make it clear where goods and services are "appearing" and where wealth (new goods and services in the market) is generated.

It is helpful to assume, in these use cases, the full utilization of legacy imagery and photogrammetry tools. The three scenarios "work" only when lots of interfaces are implemented on top of "hard, difficult" software. Much of this software is mature, but it has never been tested working together, in real time, in a seamless multi-vendor environment, because the interoperability interfaces, based on the OpenGIS Specification, are only now beginning to appear. The key to interoperability in photogrammetry and remote sensing, and also the key to "plug and play" (commodity) software components in these fields, is consensus in the marketplace at the level of software interfaces. The interfaces must be carefully specified by the industry, and then skillfully applied by software vendors, so that the resulting interoperable photogrammetric software components will be able to thrive in the marketplace. Notice that the industry consensus has to be global for the scenario to work optimally.

We need to nurture emerging commodity software components. The first commodity goods will probably be the implementations of services near the bottom of Figure 1. Excellent implementations here will give their vendors access to the larger marketplace, and generate "reputation" for their creators.

2 THE OPEN GIS CONSORTIUM, INC.

The Open GIS Consortium is a not-for-profit organization that works toward market consensus on interfaces that enable interoperability between geoprocessing systems. OGC's organization is hierarchical to keep it focused and working efficiently. The OGC Board of Directors are eminent individuals, mostly in the field of information technology, who are not necessarily employed by member organizations. Their role is to set OGC's strategic direction, maintain its bylaws, and approve its business plan. The OGC Management Committee (mainly OGC's Principal Members and standards liaisons) coordinates the business and marketing issues that naturally arise in the formation of a new sector of the economy. The OGC Technical Committee (attended by representatives from all member companies and organizations) develops interface specifications that enable the interoperation and componentization of GIS services. OGC's staff members organize the meetings, recruit new members, promote the consortium's work, and work to maintain communication within the consortium. Staff also manage the testing of commercial software that is submitted for conformance testing to ensure that the software's interfaces conform to OpenGIS Implementation Specifications.

3 OGC'S TECHNOLOGY DEVELOPMENT PROCESS

The Technical Committee's Core Task Force focuses on interfaces that will be useful across all application domains. The Domain Task Force focuses on interfaces that will benefit particular application domains, such as Telecommunications, Transportation, or Defense and Intelligence. The Revision Task Force deals with maintenance and revision of specifications.

Core Task Force Special Interest Groups (SIGs) develop white papers which describe the SIGs' purpose and general scope of work. The Image Exploitation SIG has a special focus on imaging issues, but the D&I SIG, the Feature SIG, the Metadata SIG, and others also work on specific imaging issues that relate to the topics of these SIGs. SIGs develop "use cases" which are distilled into UML-defined interfaces which become part of the OpenGIS Abstract Specification, a high level specification which is independent of computing platforms. When a portion of the OpenGIS Abstract Specification is sufficiently defined, a Working Group is created to write a public Request for Proposals (RFP) for an "implementation specification." In response to the RFP, vendors or teams of vendors submit proposed OpenGIS Implementation Specifications which are refined and approved to become industry standard interface specifications. These are engineering specifications for software interfaces which enable interoperability (on a particular computing platform) between systems that have such interfaces.

Requests for Information (RFIs) and Requests for Comment (RFCs) sometimes play a role. An RFI is a broad public announcement by OGC that a certain specification is being developed, and the RFI invites submission of information that might enable creation of a better specification. An RFC is an unsolicited submission of a proposed specification to the Technical Committee for consideration and possible adoption by OGC.

The OGC Technical Committee published its first RFP, for "Simple Feature Access and Manipulation" on September 11, 1997. As of June 23, 1998, OGC's conformance testing approach is nearing completion. Vendor implementations of OpenGIS Simple Features Specification conformant interfaces are under development or have completed and are awaiting conformance testing. After a product passes a conformance test, the vendor will receive a license to advertise the product as "conformant with the OpenGIS Simple Features Specification for SQL," or for OLE/COM, or CORBA.

OGC's second and third RFPs are for Catalog Services and Grid Coverages, both topics of interest to the photogrammetry and remote sensing community.

- The OpenGIS Catalog Services Specification will specify a common set of interfaces and services for discovery of and access to both geographic information and geoprocessing resources. Among other things, this will enable "spatial search engines," similar in concept to today's Web search engines, but able to search on location and geographic information theme.

- The OpenGIS Grid Coverages Specification ("Simple Coverages") will provide access to data such as grid-cell and raster data, images, digital terrain models, and scanned maps which are held in heterogeneous systems. The specification resulting from this RFP may include basic image manipulation techniques such as radiometric correction, contrast enhancement, noise removal, statistical calculation and histogram calculation which are commonly used in enhancing and analyzing Earth images. Note that OGC does not seek to standardize the implementation functions themselves, but rather seeks to standardize the interfaces that enable diverse clients and servers to request and respond to requests for these kinds of functions.

4 SOFTWARE ENGINEERING TOOLS IN OGC

How do infrastructure component developers endow their systems with interoperability? The answer is the great software success story of the past decade: through the use of Object Technology, and specifically through careful attention to the specification of object classes, and to the specification of interfaces on objects. Interfaces are like magic words; when a magic word is uttered in front of an interface-enabled object, it evokes a specific behavior. Interoperability is achieved by market-wide consensus on the object classes and on the names of and on the behaviors of interfaces on those classes.

For the past three years, OGC has been following the guidelines of Steve Cook and John Daniels [1] to establish a clear pathway to the definition of interfaces. They define three separate and distinct steps that lead to stable and sturdy interfaces for information systems. The first step identified by Cook and Daniels is to create a model of the world as one would like it to be. The result is called the Essential Model. The second and third steps both create models of software. The difference between them is that the purpose of the second step (called the Specification Model) is to state what the software will do, while the purpose of the third step (called the Implementation Model by Cook and Daniels -- OGC uses the term Implementation Specification) is to describe the objects in the executing software and how they communicate. Special attention is given to the definition of interfaces in the Implementation Specification.

The Essential Model and the Specification Model together are called the Abstract Specification in the Open GIS Consortium. These specifications form the "requirements" statement for the Consortium's Requests For Proposals (RFPs). The Implementation Specification is proposed by interested members of the Consortium in response to the RFP. Implementation Specifications (or “engineering specifications”) are the basis for industry consensus and, ultimately, interoperability.

4.1 The Role of Software Modeling Tools

The purpose of the Essential Model is to state the way things ought to work in plain language. Tools to assist the Essential Modeling process are most valuable. The business of creating an Essential Model usually comes down to creating use cases, finding object classes, and discovering object interactions. None of these activities is easy. Using a modeling tool to assist with the Essential Model changes things in several fundamental ways. First, the tool may bring with it a useful language for the expression of the model. Second, the tool may provide a visual modeling environment where objects, behaviors, and relationships become visual and available to drag and drop modifications. Third, the tool may automatically create objects from the visual model and also automate much of the checking of these systems of objects.

An increasingly popular software engineering tool in OGC is Rose®, (from Rational Software Corporation, Cupertino, CA) which supports a number of modeling languages. The favored language is Universal Modeling Language, or UML. UML has most of the advantages of its predecessors, which include the notations of OMT, Objectory, Booch, and a number of others [2]. It carries the additional advantage of being the modeling language of choice of ISO TC/211. Rational Rose empowers the system designer to interactively create flows of events within use cases, including preconditions, subflows, and alternative flows, and to link these flows of events to use cases. The real power comes from the visualization tool in Rational Rose. For use cases, the tool creates use case diagrams, where relationships between actors, use cases and other use cases is graphically represented [3].

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

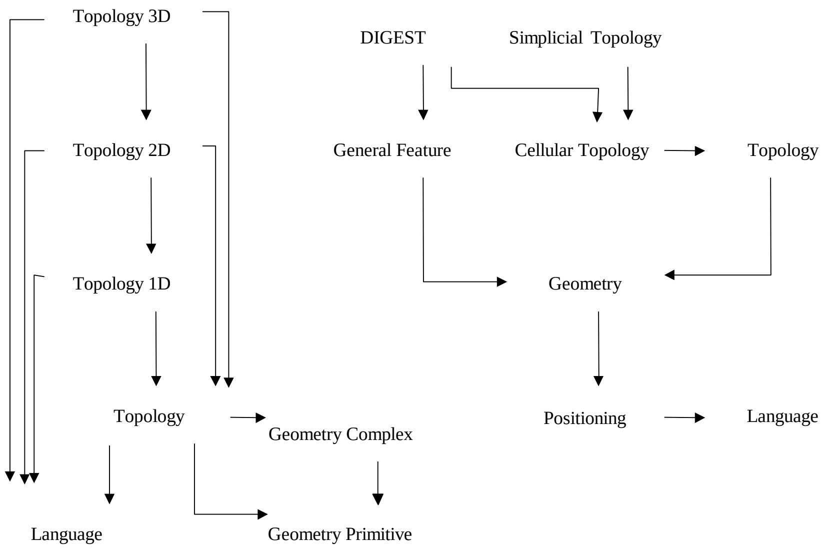

Figure 2. Two Possible Folder Diagrams

New actors, new flows, and new use cases can be created with drag and drop ease. It is much better than the old way (pencil and paper) for at least two reasons: One is the ability to share essential model information in a formal and structured manner among a team of developers (using attachments to email, for example.) Another is that the structure of the use cases are formally and symbolically stored in the system where they can be used later to verify that candidate Abstract Specifications actually accomplish their intended functions.

4.2 Impact of Modeling Tools on the Essential Specification

There are many objects and interfaces within the scope of GIS, and perhaps too many to deal with as a whole. For this reason, OGC has broken the overall GIS model into pieces, called Topics, and OGC treats each Topic separately in its own "book," or Topic Volume. There are currently 14 Topic Volumes:

1. Feature Geometry
2. Spatial Reference Systems
3. Locational Geometry
4. Stored Functions/Interpolation
5. OpenGIS Feature and Feature Collections
6. The Coverage
7. The Earth Image
8. Feature Relationships
9. Quality
10. Transfer Technology
11. Metadata
12. Services Architecture
13. Catalog Service
14. Semantics and Information Communities

A recent version of these volumes may be found at http://www.opengis.org .

Section 2 of each volume holds the Essential Model for the portion of GIS within its scope. Today, all of these sections are written in natural language (English), and none have been formally structured with a tool such as Rational Rose. Over the next year or so, especially as the Consortium moves to address the subject of conformance testing, one may expect to see a gradual rewriting of these specifications into the constructs of a formal tool such as Rational Rose.

The character of OGC's future specifications will be influenced by the tools and conventions that evolve in the field of software engineering.

The folder is an important concept in Rational Rose (see below). Universal Modeling Language (UML) has a similar concept called "packages." The folder is designed to bridge the gap between top-down functional design and bottom up object/interface design. The folder serves to group together classes and their interfaces into higher-level units where functionality may be perceived. For example, consider the two cascades of folders in the figure below. (Note that in UML, the little tab on top of a rectangle is supposed to make it look like a folder, or package.)

It must be emphasized that the folder diagrams in this figure are conceptual only. The modeling of folders in OGC has not yet progressed to the consensus process. In the hypothetical cascade of folders on the left, the top folder contains object classes that are needed for three dimensional topology. The classes in this folder are dependent upon lower level folders, as indicated by the arrows. At the bottom are folders containing primitive elements that all of the folders in that diagram need. A similar set of dependency relationships are suggested in the right-hand of the figure.

The similarities and differences in the two diagrams suggests that there is some flexibility in the design of folder diagrams, just as there is obvious flexibility in deciding exactly how many folders to use, and what classes to put in each one.

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

The impact of modeling tools upon the OpenGIS process is just beginning to be felt, but it will be extensive. Here are a few predictions that are based on the assumption that computer application software engineering tools such as Rational Rose will become more pervasive and more powerful.

1) Folders, and the folder hierarchy, suggest how conformance testing might be accomplished. Here is one scenario that employs folder-based logic:

First, implementations of object classes in the bottom layer of the folder diagram will be tested. These tests will be manual, as there is no lower level of automation to assist the testing process. Second, implementations of object classes at the next lowest level may be tested. The execution of these tests will generate service requests from lower level implementations. These service requests will be serviced by packages already (manually) tested to be conformant. The process continues forever, as implementations of higher and higher level folders (provided services to narrower and narrower niches) are tested for conformance.

2) A second prediction may be based on the fact that folders provide a mechanism to align top-down functional decomposition and service architecture design with bottom-up interface definition and object class development. In GIS, there are many functional areas, and an enormous number of services are required to support them. The mapping between low-level interfaces to the functional areas enabled by them has been exceedingly complex. The view of the relationships between functional areas and interfaces that is provided by folder technology promises to be much more transparent than previous views. The result will be a much more powerful way to prioritize interface development, and a much better estimation of the schedule and cost for component development.

3) A third prediction is based on the process just described. The mapping between interfaces and functionality revealed by the folder concept suggests how new OpenGIS conformant software components and new functionality may enter the open marketplace. Many market niches become apparent by studying the hierarchical structure in the folder diagrams. It may not make sense, for example, at least initially, for there to be more than two or three implementations of the lowest level folders. Why should this wheel be re-invented repeatedly? (When the market matures, there may be need for additional special implementations of low level folders that are especially attentive to speed, accuracy, or other qualities. But that is beside the point.) The alternative to creating custom low level folder implementations is to build implementations of higher level folders specific to targeted markets, and to exploit the implementation of lower level folders already in the market. This approach rewards partnering, provides real value to the marketplace, and assures the destruction of yesterday's closed systems. The new market promises to be exciting and beneficial to vendor and user alike. "Shrink-wrap" software components that carry the OGC certification mark will be available from many vendors for use in all kinds of applications.

5 FUTURE DIRECTIONS FOR OGC

Candidate objects of photogrammetry to be developed in OGC include: images and other grid coverages, stored functions, evaluators, mono- and stereoscopic photogrammetric models, accuracy and quality models, attributes and their values, metadata, photogrammetric processes, and their attributes and associations.

This is, of course, a very large project, and it will take some years. For example, below we outline the component elements involved in stored functions and quality models.

1) "Stored function" objects enable the communication of photogrammetric equations between clients and servers in a distributed environment, even potentially between software components that were developed without knowledge of each other. Stored functions are the stuff of OGC's Topic 4, Stored Functions, but in fact, the most advanced examples are actually in Topic 7, Earth Imagery. (This is because the Technical Committee has not finished abstracting the role of a stored function out of Topic 7 and placing it in Topic 4 where it belongs.)

Today's stored functions are tightly coupled to the specific application. For example:

- Stored functions for image rubber sheet warping
- Stored functions for single image resection
- Stored functions for map projections
- Stored functions for maintaining a stereo view while roaming

All of these are tightly coupled to their use. One can look at the function and guess the application. The result is "custom" stored functions that are proprietary, and that have no mechanism to become open and mainstream. We need to abstract from this "custom" view of functions to a more general and open view. That is, we should not be afraid to introduce a class of simple functions such as:

- Piecewise linear functions over a cellular decomposition
- Polynomial functions
- Rational functions

and use these simple ones to model the "custom" ones. The process can be as accurate as one wants by subdividing the cell structure, increasing the degree of the polynomial, etc. (One must be careful when doing this, but the elements of such care are well-known.) The idea is to have an interface built on simple functions that are exposed. Privately, implementations will still have their complex functions inside them, but these will be proprietary and not exposed. The OpenGIS "simple stored function" approach might be considered wasteful of storage and bandwidth, but it offers greater simplicity and implementability.

2) With regard to quality, we have the idea of "binning", that is, the placement of data into "quality bins" so the level of trust it carries is obvious. Of course this requires consensus on: the notion of bins; the bin thresholds and names; the interface that exposes the bin name for each data element, etc. There are good examples in OGC's Topic 9, Quality.

In the OpenGIS Specification, special attention is given to the methods (or "behaviors") supported by each object class that enable interoperability. The OGC approach

D. Fritsch, M. Englich & M. Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS - Between Visions and Applications, Stuttgart, Germany.

exploits the fact that images and other photogrammetric objects are subclasses of "GIS Feature," and so inherit many useful properties and behaviors from this and other more abstract classes. Because of this application of object modeling, the "what bin are you in?" interface would work for images, features, coverages, and all their subtypes. This enormously simplifies the design of implementations that need to interoperate. True, the notion of "bin" is somewhat crude and inexact, but nothing prevents the introduction of additional interfaces that carry more precise quality information. When these are needed, they can be added. It is likely that "bins" will satisfy 95% of the market need for accuracy and quality information.

6 REFERENCES

[1] Cook, Steve and John Daniels, Designing Object Systems (Object-Oriented Modelling with Syntropy, Prentice Hall, New York, 1994.

[2] Fowler, Martin with Kendall Scott, UML Distilled, Addison Wesley, Reading, Massachusetts, 1997.

[3]Visual Modelling with Rational Rose and UML, Terry Quatrani, Addison Wesley, Reading Massachusetts, 1996.