Open GIS Consortium

Technology Office

4899 North Old State Road 37
Bloomington, IN 47408

Telephone: +1-812-334-0601
Facsimile: +1-812-334-0625

Request For Quotation

And

Call For Participation

In the

OGC Emergency Mapping Symbology (EMS-1) Initiative

Annex B

Architecture

RFQ Issuance Date: 19 November 2003

Proposal Due Date: 16 December 2003

TABLE OF CONTENTS

Annex B: EMS-1 Architecture

1 Overview

The EMS-1 initiative attempts to fulfill part of the modernization goals of its sponsors – to increase capabilities to leverage existing market driven, Standards–based Commercial Off-The-Shelf (SCOTS) solutions for fulfilling their missions. Part of the mission is to ensure that commercial industry addresses interoperable technology requirements. EMS-1 will facilitate OGC members and industry vendors to develop, test and validate interface specifications, which are anticipated to lead to commercial products suitable for use by the sponsors, their customers and the broader geospatial community, including international, federal, state and local agencies involved in emergency management and response activities.

The EMS-1 initiative will produce an implementation of Style Management Services (SMS) that were first described and demonstrated in the Open Web Services Phase 1.2 (OWS-1.2) test bed initiative. While SMS can be thought of as a type of service, it is perhaps more practically considered a logical composition of design-patterns, service interfaces and encodings. As such, SMS defines an architecture for enabling scalable and interoperable management of symbols and styles in support of cartographic portrayal processes. The key elements of the SMS are:

- **Symbol:** a set of predefined graphical representation parameters and/or fixed graphic icons; the instructions for how vector graphics are to be represented (e.g., geometry/graphic, fill, color, stroke, font, orientation, size, opacity, etc.); the instructions for how raster graphics are to be represented (e.g., opacity, R/G/B channel selection, color map, shaded relief, contrast enhancements, etc.). HSWG[1] and GeoSym[2] are specifications defining sets of symbols for cartographically portraying features.

- **Style:** maps feature types, properties and constraints to one or more parameterized symbols; also the properties and rules describing how features are drawn during a graphical rendering process (e.g., order of layers, associate symbol type X with feature type Y, or how to apply one or more symbols to drawing a road at its centerline, etc.); **Styled Layer Descriptor (SLD)** is the XML language for defining rules for styling features and coverages.

- **Feature**: objects/phenomena on the Earth that are normally represented as graphical entities on a map (e.g., a house, political boundary, lake).

- **Feature Type** – identifies the semantic, structure (properties and property types) and behavior of Feature instances and can be defined with a GML Application Schema.

- **Coverage** – a feature that associates positions within a bounded space (its spatio-temporal domain) to feature attribute values (its range) (e.g., a digital terrain model or image)

- **Registry Information Model (RIM):** the information model that provides the means to package, publish and discover feature, style and symbol metadata by Catalog Service- Web Profile (formerly Web Registry Service).

---

[1] Symbols for Emergency Management and First Responder communities – a map symbology set under development by the FGDC Homeland Security Work Group (HSWG). See: http://www.fgdc.gov/publications/homeland.html and http://www.fgdc.gov/HSWG/

[2] Geospatial Symbols (GeoSym) for Digital Displays. The map symbology set defined by NIMA to portray Vector Product Format (VPF) data. See: http://www.nima.mil/ast/fm/acq/mil89045.pdf

- **Web Map Service (WMS):** a service that uses SLD to generate cartographic portrayals of features on the Web.

- **Web Object Service (WOS):** the means to store and access Style and Symbol instances in repositories on the Web.

- **Web Registry Service (WRS):** the means to record instances of service, feature type, symbol and style metadata for discovery and access on the Web.

Note: The Catalog Services Revision Work Group of the OGC Technical Committee is developing but has not yet released a new revision of the Catalog Services Implementation Specification 2.0 that will incorporate, as a profile, a "stateless" Web interface called CS-W that is derived from earlier work on WRS including implementations developed and demonstrated in previous Interoperability Program initiatives. So in addition to the Catalog Service 1.1.1 Implementation Specification, the WRS Interoperability Program Report (Project Document 03-024) listed in Appendix A will comprise the technical baseline for Catalog/Registry services for this initiative. The terms Registry, Catalog, WRS, CS-W and RIM are used interchangeably throughout this document to refer to this body of work.

The architecture presented herein is intended to be a starting point. This document should be considered a draft and extensions and modifications to this architecture will be generated from lessons learned through the EMS-1 and other OGC initiatives.

1.1 Critical Infrastructure Collaborative Environment

The Critical Infrastructure Collaborative Environment (CICE) is a prototype open distributed geoprocessing environment based on open architectures enabling publishing, discovery, and use of geospatial information for Critical Infrastructure Protection and other activities. CICE leverages OGC Web Services (OWS) to enable:

1) The publication of the availability of critical infrastructure services and data
2) The registration and categorization of published service and data providers
3) The discovery and use of needed critical infrastructure services and data

CICE will also establish a leave-behind capability of services, data, applications, partners, relationships, use cases, test cases, and scenarios (see CICE Architecture references listed in Appendix A).

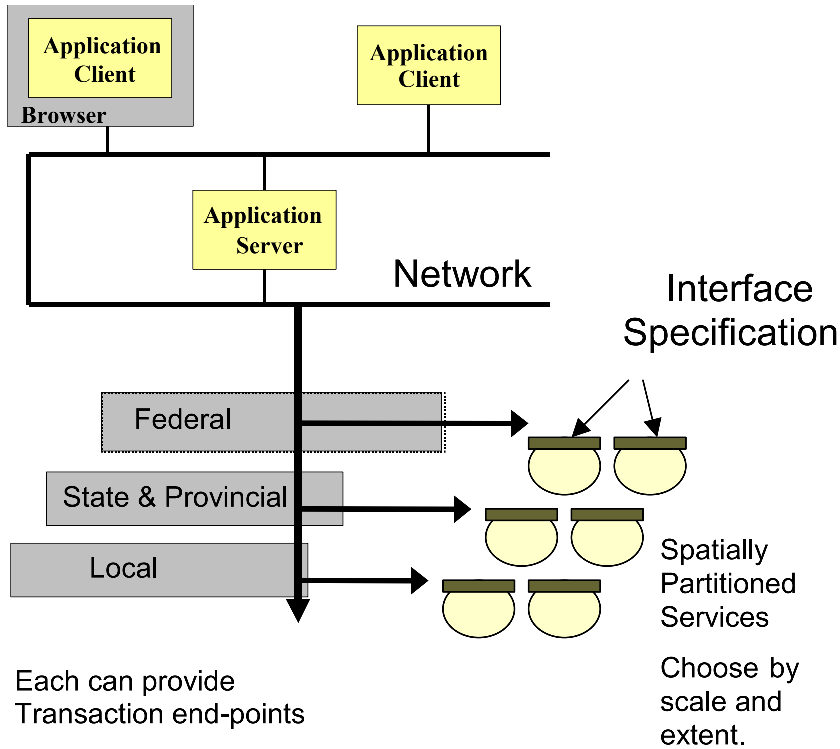
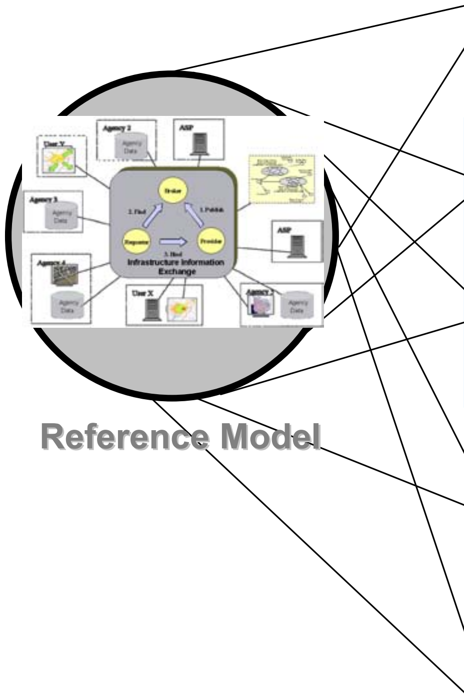


Figure 1-1 - Critical Infrastructure Collaborative Environment (CICE)

1.2 EMS-1 Reference Model

The structure of this Architecture is loosely based on the Reference Model for Open Distributed Processing (RM-ODP). The five views into the EMS-1 Architecture are described in further detail in Sections 2 through 7 of this annex.



Enterprise viewpoint: articulates a "business model" that should be understandable by all stakeholders; focuses on purpose, scope, operational objectives, policies, enterprise objects, etc

Information viewpoint: focuses on information content and system behavior (i.e. data models, semantics, schemas).

Computational viewpoint: captures components, interfaces, interactions and constraints without regard to distribution.

Engineering viewpoint: describes infrastructure and mechanisms for component distribution, distribution transparency and constraints, and binding and interaction.

Technology viewpoint: defines implementation and deployment environment using technologies, standards and products of the day.

Figure 1-2. EMS-1 Reference Model

- Section 2 (Enterprise view) describes the Enterprise Architecture for the EMS-1. This architecture describes the high-level system concept and presents representative use cases.
- Section 3 (Information View) describes the Information Architecture for the EMS-1. This architecture describes the basic information building blocks of EMS-1.
- Section 4 (Computational View) describes the Computational Architecture for the EMS-1. This architecture describes the basic service building blocks of EMS-1.
- Section 5 (Engineering View) describes the Engineering architecture for the EMS-1. This architecture describes the core components that are to be deployed and the infrastructure to integrate them into a single environment.
- Section 6(Technology View) describes the target deployment environment for EMS-1 components in terms of technologies, standards and products.

Appendix A contains references cited in Annex B

2 Enterprise Viewpoint

The EMS-1 Enterprise Architecture captures the capabilities that must be present in support of emergency management and response operations. The capabilities identified in the enterprise view provide the requirements to be met by the EMS-1 services and information architecture. Success of the EMS-1 is measured by the accuracy of the enterprise view and how well the information and systems architectures support that view.

The Enterprise Architecture is defined by a high-level system concept and use-cases. The system concept illustrates the operational setting, major system components, and major interfaces. The use cases provide descriptions of the behavior of the system from the point of view of users.

Figure 2-1 depicts the basic components of the EMS-1 concept. Data sources for emergency management are provisioned as Web-accessible services by many different, geographically distributed organizations. The organizations providing data for Emergency Management and Response are “vertically” and “horizontally” integrated across public and private sector organizations at international, national, regional, state and local levels. In the EMS-1 architecture, these data sources are published and accessible via OpenGIS® Web Service (OWS) interoperability specifications such as GML, SLD, WMS, WFS, WRS, WOS described in this and other supporting documents.

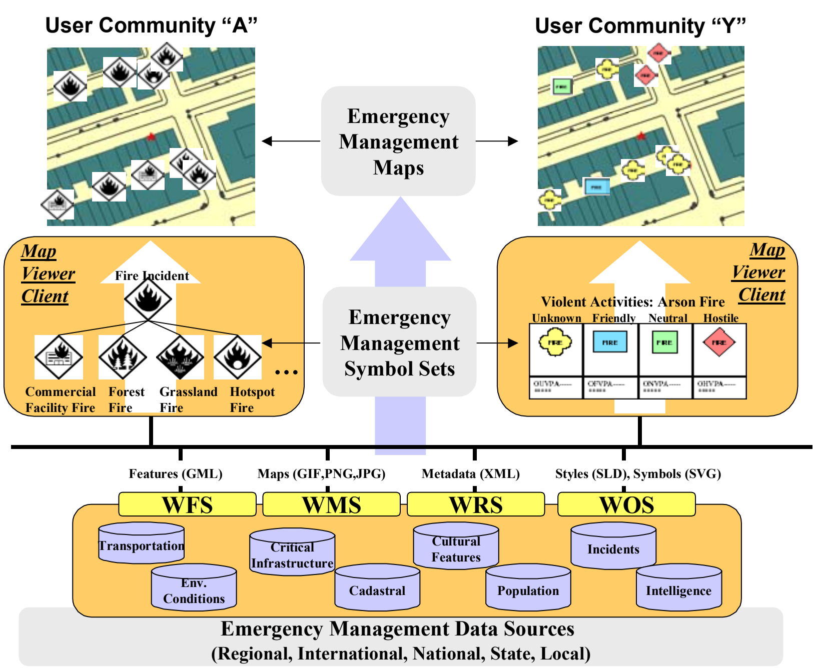


Figure 2-1 – EMS-1 Operational Context

In Figure 2-1, people (e.g., first responders and emergency management personnel) use “Map Viewer Clients” to dynamically generate and view maps of emergency incidents, critical infrastructure and other related information. The users of the system may, however, represent a wide range of organizations and “information communities” engaged in different emergency management activities including support for emergency detection, preparedness, prevention, protection, response and recovery. While all communities may use the same sources and standards for accessing geographic data, each user-community may have specialized rules for cartographically representing emergency-related information on maps. In many cases, the disparate user-communities may mandate use of styling rules and symbol sets that are designed specifically for generating map products tuned for their intended use in supporting the specialized mission of their users. Thus, in Figure 2-1, the users in "User-Community Y" (e.g., first responders) may be accustomed to viewing maps with incident symbols presented one way (presumably meaningful to their mission) and users in "User-Community A" (e.g., incident recovery planners) another way.

The challenge presented above can be mitigated somewhat to the degree that cartographic styling rules and symbol sets can be standardized and adopted across the user-communities. Nevertheless, there will always be information communities with different (possibly contradictory) requirements for map portrayal that result in the definition and use of different styling rules and symbol sets for map production. The EMS-1 Architecture is intended to enable interoperability, flexibility and re-use through a common framework of service interfaces and encodings for styles, symbols and associated metadata.

2.1 Problems and Objectives

Table 1 – EMS-1 Problem & Solution Summary

Problem Statement	Description of solution and solution inputs, outputs, constraints, variables and alternatives
1. Lack of access to relevant/appropriate/current geospatial information products. A standard solution is sought.	• A system that provides transparent information discovery and access of information based on stated information needs/requirements. • A system that enables data to be transparently (from user perspective) discovered and then accessed (via standard service interfaces and data encodings) regardless of their physical location on networks (local or wide-area networks, private or open networks). • Portable and interoperable across different service and repository implementations (i.e., SCOTS-based) • Scalable
2. Lack of interoperability of geodata (formats and semantics), symbols and services for portrayal. A standard solution is sought.	• A system that encodes disparate but commonly-used geospatial "data products" according to community/application/product-specific schemas using a common markup language (e.g., GML, SLD) • A system that defines and uses common mechanisms for defining and describing basic information types, messages, interfaces and services. • A system that defines and uses common types, messages and interfaces and ways for describing services to enable service connectivity, composition and interoperability. • A system that can use metadata (including typing information) to relate or associate symbols and styles to suit stated information needs/requirements of users.
3. Inability to easily/automatically generate customized information products fusing geospatial with other sources of information about an incident. A standard solution is sought.	• A system that enables specialized products (views) of the same geospatial information (models) to be produced, defined and/or chosen and applied. • A system that can present other (non-geospatial) data along with geospatial information in a meaningful way. • A system that supports customized packaging and delivery of information that is

Problem Statement	Description of solution and solution inputs, outputs, constraints, variables and alternatives
	"tuned" for users

Table 2- EMS-1 Objectives

Objectives
1. Stabilize, test, package and demonstrate OGC "Technical Baseline". ○ Build on OWS1.2 initiative (e.g., SMS symbol/style management, registries, RIM, GML3, Web Feature Service, Web Object Service, etc.) ○ Build on GOS (TP & PI) initiatives (e.g., data modeling, portal infrastructure, etc) ○ Build on CIPI (1&2) initiatives (e.g., security, information models, alert notification, portal nodes, schema translation, GML3 schema profiles, WFS transactions, etc.)
2. Extend OGC technical baseline with tested implementations supporting automation of cartographically styled portrayals of geospatial data (features and imagery) for the emergency mapping problem domain. ○ Provide capability to configure map services with standard style and symbol types for use with datasets for critical infrastructure and emergency management. ○ Support ability to style geospatial information in a consistent way, regardless of the source of data, according to specified "information product" rules. ○ Support standards-based portrayal of vector, image and terrain data. ○ Provide ability for client applications to choose standard views/presentations of geospatial information.
3. Extend and demonstrate capabilities for enabling "interoperable information communities". ○ Provide service infrastructure for publication and discovery of standardized application schema and feature catalogs. ○ Develop a standard classification of feature types for Home Land Security, Critical Infrastructure and Emergency Management domains and make this available on the Web. ○ Develop a standard classification of style and symbol types that apply to feature types for HLS, CI, and EM domains and make this available on the Web. ○ Implement portrayal clients and services that use these published resources (classifications and services).

2.2 Functional Requirements

The requirements listed are adapted from requirements originally developed during the OWS1.2 testbed initiative and are included here to further refine the scope and basic functionality envisioned of the EMS-1 architecture.

General Requirements

1. Must allow symbols and styles to be dynamically applied (bound) to geodata independent of its source and type, according to:

    1.1. Feature type (i.e., the semantic and structure of a geographic feature)

    1.2. Metadata for symbols, styles and feature datasets

2. Must allow applications (application clients or application services) to control how geodata will be visually presented.

3. Must allow client applications to create and store styles and symbols for use (possibly by other users and applications) for cartographic portrayal of geodata.

4. Must allow metadata to be created for describing symbol, style and feature types (and collections of these) and associated with instances of these.

5. Must allow symbol, style and feature types and their instances to be discovered and accessed for use in cartographic portrayal of geospatial information.

6. Must allow a style instance definition to be retrieved by name and identifier.

7. Must allow a symbol instance to be retrieved by name and identifier.

8. Must allow users to discover well-formed style names that are available within a style repository.

9. Must allow users to discover well-formed symbol names that are available within a symbol repository.

10. Must allow authorized users to insert new style definitions into a style repository. When a style is inserted in the style repository, the style must be validated and metadata describing the new style definition must be classified and associated with the inserted style.

11. Must allow authorized users to insert new symbols into a symbol repository. When a symbol is inserted in the symbol repository, metadata describing the new symbol must be classified and associated with the inserted symbol.

12. Should allow users to delete a style definition from the style repository.

13. Should allow users to delete a symbol from the symbol repository.

14. Store style definitions as ”feature styles” (SLD <FeatureTypeStyles> elements). This style information must be described with metadata and published and classified in the style registry.

Style and Symbol Discovery Requirements

15. The system must allow users to discover styles referenced by a specified feature type. In this discovery, a user requests style names, giving the feature type name as a search parameter. The system

must respond to the request with a list of well-formed style names that apply styles for the given feature type.

16. The system must allow users to discover styles based on one or more properties defined within the style metadata and a style classification.

17. The system must allow users to discover symbols based on one or more properties defined within the symbol metadata and a symbol classification.

Style and Symbol Metadata Requirements

18. Style definitions must be annotated with metadata that describe properties of the style. These properties shall include application domain (e.g. “weather forecasting” or “command and control”) and organization domain (e.g. “NIMA” or “USGS”) identifiers.

19. Symbols definitions must be annotated with metadata that describe properties of the symbol. These properties shall include application domain (e.g. “weather forecasting” or “command and control”) and organization domain (e.g. “NIMA” or “USGS”) identifiers.

Style and Symbol Representation Requirements

20. The system must support different symbol representations (i.e., encodings, data file formats) without requiring knowledge of their representation. The system must have knowledge of the metadata representation. Therefore, the system will have the ability to work with style and symbol representations based on their metadata.

21. The system should have the capability to transform style information between representations, i.e. to provide a style in any of the formats/encodings the system supports. Considering the previous item, supporting a representation means the ability to always produce style and symbol document instances in a supported encoding.

22. The system must be able to check the validity of symbol and style documents to be stored in the repository. The validation will be performed exclusively against the schema of the document that is provided with the request together with the input data. Invalid data must be rejected. The validation will be performed only once – on input.

2.3 Architecture Requirements

1. The Architecture shall provide enough detail to show how it fits into the OGC Critical Infrastructure Collaborative Environment (CICE)
2. The Architecture shall focus on the exchange of information through online services
3. The Information Architecture shall focus on Critical Infrastructure and Emergency Mapping related geographical data
4. The Architecture shall be built on open standards for interoperability
5. There shall be two Roles that organizations will play in support of the EMS-1:
   5.1. Requestors of EMS Information
   5.2. Providers of EMS Information
6. The EMS-1 shall provide the ability for actors in those Roles to:
   6.1. Publish EMS Information
   6.2. Access EMS Information

7. Create and Update EMS Information
8. Portray maps.
9. Within the Architecture, Distributed Geoprocessing Resources shall be defined as information from multiple sources available as network addressable instances of typed data or services (OpenGIS® Web Services) including Data Services, Portrayal Services, Processing Services, and associated Encodings
10. Local, State, Federal, and private sector organizations shall be able to publish, find, access, integrate and apply Distributed Geoprocessing Resources across a collaborative network environment
11. Distributed Geoprocessing Resources shall be accessible:
    11.1. Vertically - Information sharing among Federal, Local and State departments/agencies, Non Governmental Organizations, and Private Sector Companies
    11.2. Horizontally - Information sharing between Federal, Local, or State departments/agencies, Non Governmental Organizations, and Private Sector Companies.
12. Information may also be shared on a transnational basis.
13. The EMS-1 Architecture efforts shall maintain coordination with ongoing related activities.
14. Within the EMS-1 Architecture, Application Clients shall be able to address HLS requirements between and across organizations where the data/service (transaction) end-points are at Federal, State, or Local levels.

2.4 Design Principles

No architecture is truly meaningful or valuable until it is shown to be implemented and useful. A process framework is therefore needed to support and enforce conformant implementation. This clause presents elements of a process framework, including design principles and high-level goals, constraints, assumptions and guidelines for EMS-1 design and specification. Elements include:

1. Everything is a network resource including clients, services, data content, appliances, and computers.
2. Resources (especially services) have contracts (i.e., resources must have well-defined roles, responsibilities, interfaces, and semantics)
3. Interoperability of services over time is maintained by focusing on commitment to contracts not adherence to static protocols.
4. Design for availability through dynamic discovery of resources:
    4.1. Assume networks are unreliable and will fail
    4.2. Do not assume given resources are always available in the same location
    4.3. Promote "self-healing" through dynamic discovery and fail-over to other resources
    4.4. A dynamic distributed application relies on the availability of multiple instances of any given resource type; the more effective the typing framework, the more dynamic and therefore reliable the application can become.
5. Maximize stateless behavior of resources:
    5.1. Resources are accessed on a transient basis.

5.2. Persistent state should be maintained solely on the tier that is interested in (responsible for) the specific computational transaction (this may also be termed separation of concerns)

5.3. Minimize dependencies of resources on the network; maximize loose coupling (i.e., minimize hard-coded dependencies between resources such as client to service instances).

5.4. Services should maintain state only within the context which requires it for a particular application (e.g. single request, single transaction of multiple requests with one client, single task sequence of transactions with multiple clients, etc.).

5.5. Service states which are to be maintained indefinitely (e.g. Web Map client contexts, user access profiles) should be managed as (metadata) resources in their own right rather than as stateful service behavior.

5.6. Clients and services should limit the duration over which they hold resources to minimize chances of losing the resource and maximize reuse.

6. To maximize reuse, assume resources will be deployed and used in different application and deployment contexts over time

Assume different deployment platforms and network communications protocols (e.g., transactional synchronous, point to point asynchronous messaging, broadcast asynchronous messaging, real time streams, low bandwidth formats, etc) will be required

2.5 Use Cases

The use cases in this section describe general scenarios that involve Style Management Services (SMS) in support of EMS-1. These should be viewed as context for the requirements listed above and the Information, Computation and Engineering viewpoints presented in other sections of this Annex.

These use cases have been extracted from OGC Project Document 03-030 “Style and Symbol Management Services Requirements” developed during the OWS1.2 Testbed initiative. They are included here for informational purposes.

2.5.1 Create map from single or multiple WFSs.

Priority	High.
Description	The user issues a request to a mapping system. The request specifies a list of feature type names, styles, and a particular portrayal service. An image is rendered and returned to the user.
Preconditions	The user must know the names of the features or feature types that he wants portrayed. The user must either know the names of desired styles or some other properties that uniquely identify the style.

Typical Flow of Events
1. Find a WFS. Receive WFS bindings. (The mechanisms used for finding a WFS via a Service Registry fall outside the scope of this use-case. Refer to WRS references in Appendix A.) 2. Issue a request to a Style Registry to obtain a style. (This step encompasses many possibilities, as

enumerated in the detailed use cases). 3. Find a portrayal service. Receive bindings. (The mechanisms used for finding a portrayal service fall outside the scope of this use-case. Refer to WRS references in Appendix A.) 4. Issue a request to the portrayal service (e.g., a WMS) that includes the styles
Alternative Paths
1. User receives a “no such WFS” exception and ends the task. 2. User receives a “no such style” exception and ends the task. 3. User receives a “no such portrayal service” exception and ends the task. 4. User receives an “unsupported operation” exception if the portrayal service is unable to render the given style.

Postconditions
Unless exceptions are raised, a raster (or possibly vector) depiction of a map is returned.

2.5.2 Using SLD to Portray a Coverage

Priority	High.
Description	The user wishes to use a Coverage Portrayal Service using SLD to style the coverages.
Preconditions	There must exist a WCS providing coverages to be portrayed, a CPS that can accept and interpret SLD, and a client that can construct the appropriate GetMap requests including the SLD.

Typical Flow of Events
1. A Web Coverage Server publishes a layer containing a multi-band swath of data over some region. 2. A client discovers that the published coverage overlaps the current region of interest and decides to graphically view the data. 3. The client builds an SLD that describes a mapping from raw data values to color values. 4. The client formulates a WMS request that includes the SLD and sends it to the CPS. 5. The CPS receives the request, fetches the coverage and/or SLD, renders the result, and returns a raster to the client.
Alternative Paths

None.

Postconditions
The result is a visual portrayal of the coverage that can be overlaid with additional WMS map layers.

2.5.3 Browsing Available Styles

Priority	Medium.
Description	The user wishes to browse the styles available from a given server.
Preconditions	None.

Typical Flow of Events
1. User issues a request for a list of available styles, possibly grouped together by some attribute of the style (such as feature type, symbology set, or application domain). 2. User receives and parses the list. 3. User (repeatedly) issues a request for a "legend graphic" or "preview image" corresponding to a given style in the list. 4. User selects a style and issues a request for the corresponding style document.
Alternative Paths
None.

Postconditions
An SLD document or other style definition is returned to a user based on the user's request.

2.5.4 Additional detailed use-cases.

Refer to the SMS Requirements Interoperability Program Report (OGC Project Document 03-030) for the remaining detailed use cases whose names are listed here:

1. Requesting Styles from a Style Management Server

   1.1. Retrieve style by name

  1.2. Find Style by Feature Type Name List
  1.3. Find Style by Style Metadata
  1.4. Find Style by Symbols Used
  1.5. Find Style by Combination of Criteria
  1.6. Dynamic Creation of SLDs
2. Pushing Information Back to a Style Management Service
  2.1. Inserting a Style into a Style Management Service
  2.2. Inserting a Symbol into a Style Management Service
  2.3. Deleting a Style into a Style Management Service
  2.4. Deleting a Symbol from a Style Management Service
  2.5. Combining Multiple Styles
3. Symbol Management
  3.1. Find Symbol by Classification
  3.2. Find Symbol by Property
4. Styles as they Relate to Other OGC Services
  4.1. Query a Portrayal Service for Styling Capabilities

3 Information Viewpoint

The information viewpoint is concerned with the information processing semantics, information system constructs (information model, semantics, data models, schemas), concepts, rules and structures of the architecture independent of distribution and implementation details.

Digital symbolization standards that support multiple portrayals of geospatial data are required to facilitate interoperable map analysis and data sharing. There are many ways to graphically portray geospatial information. Geographic features, for example, may have multiple symbols assigned to them. The choice of which symbol to apply to a feature for portrayal may need to be made dynamically, depending on its type, values of its properties, the application in which it is portrayed and the preferences of the user viewing it. There is a need to portray the same feature in different ways. The Style Management Services (SMS) architecture is designed to enable more flexible and interoperable map portrayal.

The SMS information architecture (Figure 3-1) has two major components that can function independently: a Style Manager and a Symbol Manager. Each of these components is comprised of two subcomponents: a Catalog-Registry and a Repository. The purpose of the respective catalog objects is to manage metadata about extrinsic data representing style and symbol objects and to provide an interface for searching and discovery of these objects. The purpose of the repository objects is to store instances of style and symbol objects and to provide an interface for clients (e.g., SMS Clients) to access to them.

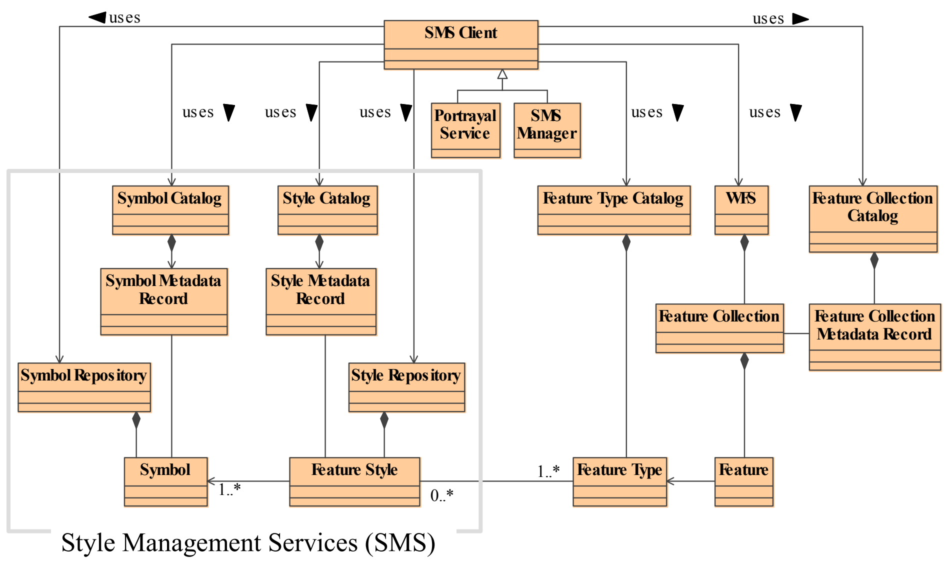


Figure 3-1. SMS Information Model

The following clauses describe the objects and their relationships depicted in the object model (Figure 3-1) adapted from the SMS Discussion Paper (OGC Project Document 03-031).

3.1 SMS Client

An SMS Client object is a consumer of SMS (Symbol Catalog, Symbol Repository, Style Catalog, Style Repository) object services. There are two kinds:

- **Portrayal Service –** provide visualization of geospatial information. Portrayal Services are components that, given one or more inputs, produce rendered outputs (e.g., cartographically portrayed maps, perspective views of terrain, annotated images, views of dynamically changing features in space and time, etc.).

- **SMS Manager –** augment geospatial information by allowing style and symbol objects and their metadata to be created, read, updated and deleted. Supports ability to manage associations between Feature Type and Feature Style objects and Feature Style and Symbol objects. Allows users to preview the appearance of symbol and style instance objects bound to feature objects.

3.2 Symbol Catalog

Provides the basic mechanism for SMS Client objects to publish and discover essential operational information about Symbol Objects. Allows the ability to manage (create, read, update and delete) and search for Symbol Metadata Records.

3.3 Symbol Metadata Record

Symbol Metadata Record is an object describing Symbol Objects. The metadata requirements of Symbols are analogous to those of Style Objects.

3.4 Symbol Repository

Allows the ability to create, read, update and delete Symbol Object instances in a repository. Allows SMS Client objects to publish and access Symbol Object instances.

3.5 Symbol Object

Symbols are pieces of graphics used by Portrayal Services to represent geographic features on a map. Symbols are the instructions for how vector and raster graphics are to be portrayed. Vector graphic symbols may have properties such as geometry, graphic, fill, color, stroke, font, orientation, size, opacity etc. Raster graphic symbols may have properties such as opacity, RGB channel selection, color map, shaded relief, contrast enhancements, etc.

Symbols may be represented as a set of graphical instructions using encoding languages such as SLD, SVG or CGM or as a raster graphic (image file) encoded in one of a set of standard formats (e.g., JPG, PNG, GIF, CGM). In principle, any number of Symbol representations may be used with the SMS architecture.

3.6 Style Catalog

Provides the basic mechanism for SMS Client objects to publish and discover essential operational information about Style Objects. Allows the ability to manage (create, read, update and delete) and search for Style Metadata Records.

3.7 Style Metadata Record

A Style Metadata Record is an object describing Style Objects. The metadata includes descriptive information such as titles, keywords and contact information (such as defined in ISO 19115/19117/19119 specifications).

Each Style Object must have a name that can be used as a primary key for identification and access. In addition to referencing this style name, Style Metadata Record objects may also contain references for all of the Feature Types to which a style can be applied. This allows a SMS Client, after locating feature styles, to more easily locate and access Style Objects

3.8 Style Repository

Allows the ability to create, read, update and delete Style Object instances in a repository. Allows SMS Client objects to publish and access Style Object instances.

3.9 Feature Style Object

A Feature Style Object is a description of how to portray a type of feature or coverage in a particular way. Styles are used by Portrayal Services. Styles can specify conditions that allow Features to be portrayed based on the value of a Feature Object’s type and its geometric and non-geometric attributes. Styles are composed of one or more Symbols that may be stored with the Style Object or separately managed by SMS. Associations of Symbol Objects with geographic Feature Objects are explicitly given through Styles.

In principle, any number of style representations may be used with the SMS architecture. The primary representation, however, is based on the OGC Styled Layer Descriptor Implementation Specification (SLD).

3.10 Feature Type Catalog

Provides the basic mechanism for application objects (including SMS Clients) to discover operational information about Feature Types. Allows the ability to manage (create, read, update and delete) and search for Feature Types.

Figure 3-2 shows the relationships and roles of features, feature types, catalogs (i.e., registries) and repositories. The central idea of this information viewpoint is that features (instances) of well-known type are measured (i.e., digitized) and recorded (i.e., archived) according to a Collection Process and the types of features to be collected. Feature instances and associated metadata are archived as Feature Collections. Feature Collections have metadata describing their content, use, lineage, sources and feature types.

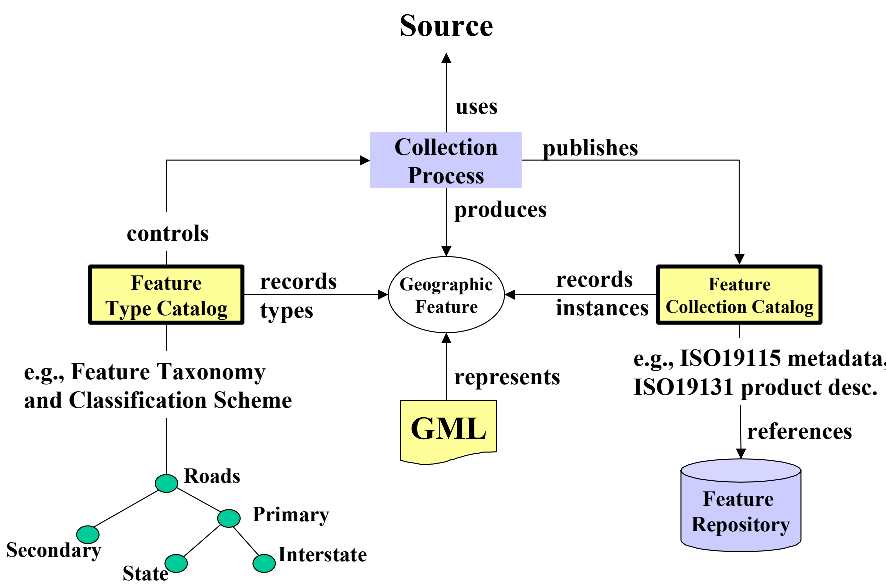


Figure 3-2. Role of Feature Taxonomies, Catalogs and Repositories

Feature instances have well-known types that are classified according to taxonomies with commonly agreed-upon semantics for a domain. Feature types are instantiated according to a schema defining a standard way of representing domain-specific types with a common set of properties, relationships and semantics (as GML). In a feature production process, an authority typically controls the feature type taxonomy and the schema. Taxonomies, schema and metadata are resources about feature types and feature instances (also resources) that are controlled, referenced and accessed in Catalogs and Repositories.

Taxonomies, schema, metadata, feature instances, feature collections, registries and archives are all resources that must be related and accessed within a “semantic web”. Catalogs of resource types and instances play a critical role in enabling semantic interoperability.

3.11 Feature Type Object

Feature Type Objects are nodes in a taxonomy of feature types. They have or reference information about the feature type taxonomy itself, schema describing the structure of feature instances of the type as well as zero or more Feature Style Objects (and their metadata) that are appropriate for use in visual portrayal of feature instances of the type. Schema defining Feature Type objects may be encoded with GML.

3.12 Web Feature Service

A service object to access Feature Object representations from Feature Collections stored in repositories.

3.13 Feature Collection

A Feature Collection is a collection of Feature Object instances. Feature Collections are themselves valid features and can have location and other properties as defined in their schema.

3.14 Feature Object

Feature Objects are instantiations of Feature Types and abstractions of real world phenomena. Representations may be encoded with GML.

3.15 Feature Collection Catalog

Provides the basic mechanism for application objects (including SMS Clients) to discover operational information about Feature Collections. Allows the ability to manage (create, read, update and delete) and search for Feature Collections. Refer to Feature Type Catalog discussion and Figure 3-2.

3.16 Feature Collection Metadata Record

A Feature Collection Metadata Record is an object describing Feature Collection Objects. The metadata includes descriptive information such as titles, keywords and contact information (such as defined in ISO 19115/19117/19119 specifications). In addition, Feature Collection Metadata Records may directly or indirectly reference the Feature Type Objects of the Feature Objects it contains. Feature Collection Metadata Records are managed by Feature Collection Catalogs.

4 Computational Viewpoint

The Information Technology environment in which the EMS-1 will take place is the Critical Infrastructure Collaborative Environment (CICE). The CICE is based on the OpenGIS® Service Framework (OSF) defined in the OpenGIS® Reference Model, and relevant elements of the US National Spatial Data Infrastructure (NSDI) and Canadian Spatial Data Infrastructure (CSDI).

The OSF describes a computational model for OpenGIS® Services. The objective of the OSF is to detail how geospatial software services plug into broader interoperability infrastructures to use and extend diverse, loosely coupled sources of data and services. The OSF draws on Topic 12 of the OGC Abstract Specification (Service Architecture-ISO 19119) but focuses more specifically on current technologies, platforms and mechanisms for enabling implementation of interoperable services.

The EMS-1 Computational Architecture provides a platform for geospatial interoperability between applications and critical infrastructure and emergency management information resources. This Framework includes the following:

- The OpenGIS® Service Framework (OSF) of the OpenGIS Reference Model establishes the basis for common service interfaces and data exchange protocols that can be utilized by any application.
- OpenGIS® Implementation Specifications provide guidance to application developers on how to build their applications to comply with this framework.
- OpenGIS® Services are implementations of services that conform to OpenGIS® Implementation Specifications.

Compliant applications, called OpenGIS® Applications, can then "plug into" the framework to join the enterprise operational environment. This loosely coupled approach to enterprise development results in very agile systems.

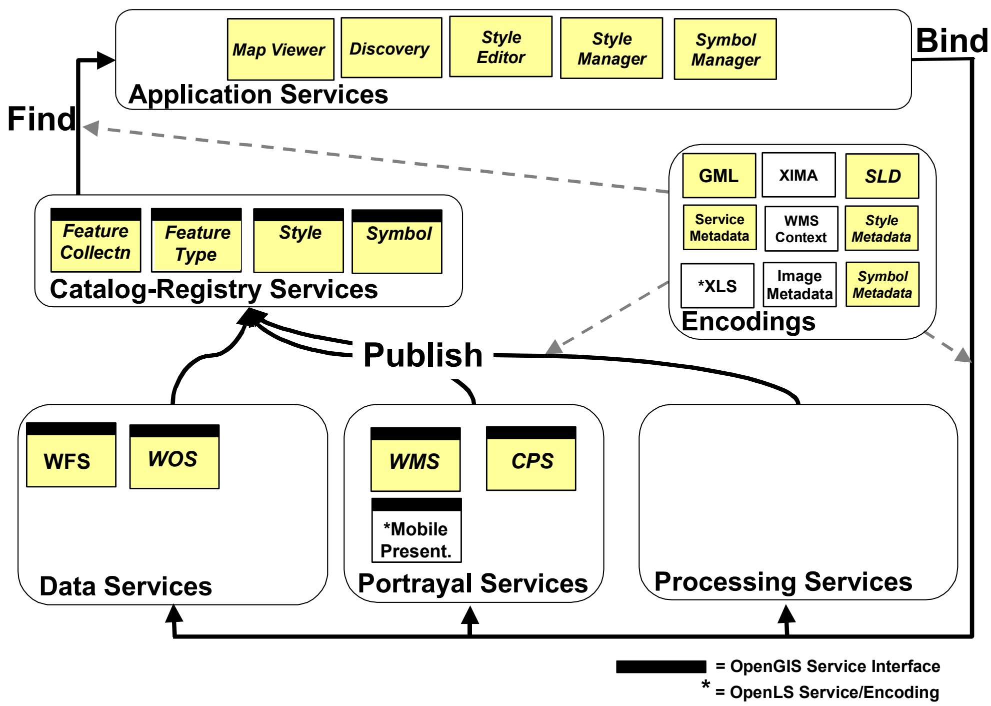


Figure 4-1. SMS Framework for EMS-1

Figure 4-1 identifies for Style Management Services Framework, a profile of the OpenGIS Service Framework, for EMS-1. All components shaded (yellow) in the above figure are key to the SMS Framework for EMS-1. Italicized components are anticipated to require enhancement based on EMS-1 requirements and previous work to enhance the specifications on which they are based. Elements of this computational framework are described in greater detail below.

4.1 Application Services

4.1.1 Web Map Viewer Client

A Web Map Viewer Client, including server-based clients, can issue GetMap requests for different maps to several independent Web Map Servers. If each map has the same geographic area and physical dimensions, and if their backgrounds are transparent, then they can be overlaid in a single window to produce a combined map. For example, server A might produce a topography image, server B a map of rivers and lakes, and server C a map of watershed boundaries. Each server maintains the type of data in which it specializes, but the end user can obtain a combined presentation of the three Layers. The Web Map Viewer Client may itself perform the portrayal process, acting as a tightly coupled Portrayal Service, or it may delegate to a loosely coupled Portrayal service, such as a WMS, to produce a map.

For EMS-1, Web Map Viewer Clients must support the ability to use the SMS components when requested by a user to generate a cartographically portrayed picture. If the Web Map Viewer Client acts as a Portrayal Service (see description below) to render maps itself, it must specifically support the EMS-1 baselines for SLD (refer to Section 6). It is anticipated this initiative will further enhance the SLD schema and other related SMS specifications as needs require. In this case, implementation of Web Map Viewer Clients for EMS-1 must be modified accordingly.

4.1.2 Value Added Client

Value-Add Clients are a class of Application Service specializing in supporting the ability for users to collect and submit user input that augments geospatial information originally supplied by a data producer. Value-Add Application Services support augmentation of data by creating new features, and updating or deleting existing features, styles, symbols and metadata. Value-Add Application Services typically support human interaction controls, the ability to add and remove layers, and the ability to create, select, and display cartographic styles to support of the value-adding process. Value-Add Application Services may also support the ability to generate preview graphics, draw on a background map and commit updates to repositories and databases using OpenGIS Data Services such as WFS and WOS.

For EMS-1, a SMS Manager client is envisioned for supporting the management of styles, symbols and metadata as required to support map production for interoperable Emergency Management and Response applications. The SMS Manager should support the ability to:

1. Discover Feature Types, Styles and Symbols published to SMS-enabled facilities (Catalogs and Repositories).

2. Create, read, update and delete Style Objects (i.e., SLD elements) to Style Repositories and associated Style Metadata Records to Style Catalogs.

3. Create, read, update and delete Symbol Objects (e.g., SVG or CGM elements) to Symbol Repositories and associated Symbol Metadata Records to Symbol Catalogs.

4. Link Feature Style Objects to Feature Type Objects and save results to SMS facilities.

5. Link Symbol Objects to Feature Style Objects and save results to SMS facilities.

6. Generate preview graphics (and/or full-fledge maps) showing the dynamic binding of Styles to Feature Objects.

4.2 Catalog-Registry Services

Catalog-Registry Services provide a common mechanism to classify, register, describe, search, maintain and access information about network resources. Resources are network addressable instances of typed data or services. Registries may be differentiated by their role such as registries for cataloging data types (e.g., types of geographic features, coverages, sensors, symbols), online data instances (e.g., datasets, repositories, symbol libraries), service types (e.g., portrayal, processing, data services) and online service instances.

The metadata content published to the registry, while conforming to the same Registry Information Model (RIM), describes different kinds of resources using metadata that may be structurally and semantically different than metadata for resources of other types or for other purposes or organizations. The OpenGIS Web Registry Service defines a common information model and the service interfaces to access resource offers, regardless of the type of resource and the content of the metadata. Refer to Appendix A for references on Registry Services and their role in the OpenGIS Service Framework.

Type Registries contain metadata about resource (data and service) types (e.g., types of images, features, feature collections, styles, symbols, and services) as taxonomies that are shared and used within information communities. The ability to publish and share this information is an essential requirement for distributed applications to be able to share and exploit, with a common semantic, these resources. Type Registry Services provide access to these metadata and taxonomies of types. Support for publishing and referencing taxonomies is explicitly supported in the Web Registry Service (WRS) Interface and Registry Information Model (RIM) specification (Appendix A).

4.2.1 Feature Type and Feature Collection Catalog-Registry

It is anticipated that both the RIM and the WRS will be used and extended (as required) to support the requirements of EMS-1. In particular, the following capabilities are desired of Catalog-Registries for Feature Types and Feature Collections:

1. Support for construction and publication of feature type taxonomies
2. Ability to publish and find schema describing the structure of feature types.
3. Ability to publish and find metadata about features and/or feature collections.
4. Ability to associate feature instances and feature collections to classification nodes in a feature taxonomy
5. Ability to bind schema to feature types and feature instances
6. Ability to bind styles to feature types
7. Ability to bind symbols to styles

4.2.2 Style and Symbol Catalog-Registry

As with the Feature Type and Feature Collection Catalog-Registries, the role of Style and Symbol Catalog-Registries is to support discovery, access and management of style (SLD) and symbol libraries. Style and Symbol Catalog-Registry Services provide access to metadata about these resources and to the published symbol and style instances accessible in repositories via Web Object Services (WOS).

The following capabilities are desired of style and symbol registries:

1. Support for construction and publication of symbol type taxonomies
2. Ability to publish and find metadata about styles and symbols.
3. Ability to associate style and symbol instances to classification nodes in a style and symbol taxonomies
4. Ability to bind styles to feature types
5. Ability to bind symbols to styles

4.3 Portrayal Services

Portrayal Services provide specialized capabilities supporting visualization of geospatial information. Portrayal Services are components that, given one or more inputs, produce rendered outputs (e.g., cartographically portrayed maps, perspective views of terrain, annotated images, views of dynamically changing features in space and time, etc.) Portrayal Services can be tightly or loosely coupled with other services such as Data and Processing Services and transform, combine, or create portrayed outputs. Portrayal Services may use styling rules specified during configuration or dynamically at runtime by Application Services.

4.3.1 Styled Layer Descriptor enabled Web Map Service (WMS)

A Web Map Server (WMS) generates "pictures" of georeferenced data. Independent of whether the underlying data are simple features (such as points, lines and polygons) or coverages (such as gridded fields), the WMS produces an image of the data that can be directly viewed in a graphical web browser or other picture-viewing software. An extension of the basic Web Map Server is the Styled Layer Descriptor (SLD) Web Map Server. The SLD enabled WMS inherits all of the attributes from the Web Map Server and adds support for the use of Styled Layer Descriptor documents to specify styling. Instead of generating maps of particular named layers in one or more predefined styles, an SLD Map Server extracts features from a data provider and renders them using a stylistic description encoded in XML.

The WMS instances used for EMS-1 must support the ability to interconnect with one or more WFS instances to access feature data and apply appropriate styles and symbols (as specified by SLD-encoded style elements associated with feature types) to produce a cartographically rendered map. For EMS-1, the WMS implementations must also support the SLD enhancements developed and tested in the OWS1.2 testbed (refer to SMS Discussion Paper, OGC project document 03-031).

4.3.2 Styled Layer Descriptor enabled Coverage Portrayal Service (CPS)

The Coverage Portrayal Service (CPS) defines a standard interface for producing visual pictures from coverage data. Typically coverage data are retrieved via a WCS instance. CPS extends the WMS interface and uses the Styled Layer Descriptor (SLD) language to support rendering of WCS coverages. CPS facilitates wider use of coverage data by making views of coverages visible within thin-clients (e.g., Web browsers). To a service requestor, the CPS appears as a WMS instance, but with additional parameters to control the retrieval and/or rendering of coverage data. The CPS may require the client to specify the targeted WCS.

CPS may be used to support:

- assigning multi-spectral bands in an image to color channels in a picture,
- creating chloropleth maps from coverage data using client-specified color-bins
- preset rendering mechanisms such as hill-shaded elevation
- combining multi-spectral pixel values according to client-specified or server-defined formulas (e.g., Normalized Difference Vegetation Index).

The CPS instances used for EMS-1 must support the ability to access coverage data and apply appropriate styles and symbols (as specified by SLD-encoded style elements associated with coverage types) to produce a cartographically rendered map. For EMS-1, the CPS implementations must also support the SLD enhancements developed and tested in the OWS1.2 testbed (refer to SMS Discussion Paper, OGC project document 03-031) and enhanced further during this initiative.

4.4 Data Services

Data Services provide access to collections of data in repositories and databases. Resources accessible by Data Services can generally be referenced by a name (identity, address, etc). Given a name, Data Services can then find the resource. Data Services usually maintain indexes to help speed up the process of finding items by name or by other attributes of the item. The OpenGIS® Framework defines common encodings and interfaces in which multiple, distributed Data Services are accessed and their contents "exposed" in a consistent manner to other major components. The sections below describe the current set of Data Services of the OpenGIS® Framework.

4.4.1 Web Feature Service (WFS)

The Web Feature Service supports the query and discovery of geographic features. In a typical Web-base scenario, Web Feature Service delivers GML (XML) representations of simple geospatial features in response to queries from HTTP clients. Clients (service requestors) access geographic feature data through a WFS by submitting a request for just those features that are needed for an application. The client generates a request and posts it to a WFS instance (a WFS server on the Web). The WFS instance executes the request, returning the results to the client (service requester) as GML. A GML-enabled client can manipulate or operate on the returned features.

For EMS-1, the WFS service must support the output of GML 2.1 encoded feature data appropriate for use in the Critical Infrastructure Protection and Emergency Management domain.

4.4.2 Web Object Service (WOS)

The Web Object Service defines a standard interface to an object repository. It defines a set of generic XML types from which object access and management services, such as WFS and WRS, may be derived. It also describes an unspecialized instantiation of the types defined with XML Schema to define a Web Object Service. Like the WFS and WRS, the WOS supports INSERT, UPDATE, DELETE, QUERY and DISCOVERY operations on object instances other than (but not excluding) GML features. Object instances may be encoded directly into a WOS request message, using XML, or they may be referenced using other mechanisms described in this document.

For EMS-1, the WOS service will be used to enable Web-based publication of and access to Symbol and Style objects held within repositories.

4.5 Encodings

4.5.1 GML

The Geography Markup Language (GML) is an XML encoding for the transport and storage of geographic information, including both the geometry and properties of geographic features. GML utilizes the OpenGIS® Abstract Specification geometry model which has been harmonized with the ISO geospatial geometry model. Unlike a simple geometry model, the GML Specification also includes the ability to handle complex properties. Work is underway to harmonize this complex property model with the ISO efforts in the same arena.

GML is used to represent geographic features conforming to well-defined application schema for purposes of transport across computational interfaces. For EMS-1, GML will be used to represent features within the Critical Infrastructure Protection and Emergency Management domain.

4.5.2 Style Layer Descriptors (SLD)

The Styled Layer Descriptor (SLD) encoding specifies the format of a map-styling language for producing georeferenced maps with user-defined styling. This language is used to create XML documents that control the visual portrayal of the data with which they work. The ability for a human or machine client to define the styling rules requires a styling language that the client and server can both understand. The SLD language can be used to portray the output of Web Map Servers, Web Feature Servers and Web Coverage Servers. The SLD is defined using XML Schema.

For EMS-1, the SLD enhancements developed and tested in the OWS1.2 testbed (refer to SMS Discussion Paper, OGC project document 03-031) will be used as a baseline for implementation and, if required, enhancement.

4.5.3 Style Metadata

Metadata for describing Style Objects. For EMS-1, these schemas were developed and tested in the OWS1.2 testbed (refer to SMS Discussion Paper, OGC project document 03-031) and provide the basis for implementation and, if required, enhancement.

4.5.4 Symbol Metadata

Metadata for describing Symbol Objects. For EMS-1, these schemas were developed and tested in the OWS1.2 testbed (refer to SMS Discussion Paper, OGC project document 03-031) and provide the basis for implementation and, if required, enhancement.

5 Engineering Viewpoint

The Engineering Viewpoint is concerned with the infrastructure required to support system distribution. It focuses on the mechanisms and functions required to: a) support distributed interaction between objects in the system and b) hide the complexities of those interactions. It exposes the distributed nature of the system, describing the infrastructure, mechanisms and functions for object distribution, distribution transparency and constraints, bindings and interactions.

Recall that Symbol Management Service (SMS) is an architecture for describing service behavior and functionality and may be implemented as a physical service at a single endpoint or a logical service implemented as a composition of interfaces at different endpoints. As with all OWS services, the binding mechanism that enables transparent service distribution is HTTP GET/POST with XML and MIME encodings used for message transport.

Figure 5-1 shows a possible deployment of SMS services on the Web. Notice that all SMS components (Symbol Catalogs and Repositories, Style Catalogs and Reposities, Feature Catalogs and Repositories) can be physically separated, replicated and even re-purposed for use in different applications at different facilities and servers. These same SMS services and the SMS Client could just as easily and transparently be physically deployed all on the same host server.

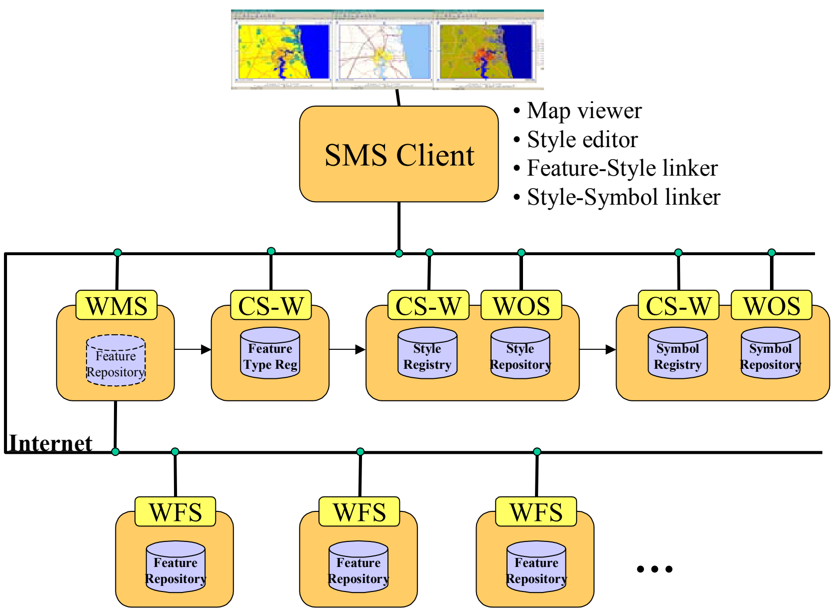


Figure 5-1. Deployment of SMS Framework for EMS on the Web

6 Technology Viewpoint

The Technology Viewpoint is concerned with the choice of technologies to support system distribution. It defines the implementation and deployment environment using technologies, standards and products of the day, providing "reference points" for conformance testing.

The following specifications are to be supported as the technical baseline for implementation of EMS-1 capabilities. These specifications may however require enhancements to achieve the EMS-1 objectives and successful implementation of requirements. In this case, software implementations must also be modified to support the enhanced specifications.

Data	
Geographic Vector Data	○ GML 3.0 – OGC Geography Markup Language 3.0. OGC Document 02-023r4. ○ Level 0 Profile of GML3 for WFS – CIPI1.2 Interoperability Program Report. OGC Document 03-003r9.
Symbol Data	○ Scalable Vector Graphic. W3C Recommendation http://www.w3.org/TR/SVG11/ ○ Computer Graphic Metafile (CGM). ISO/IEEE 8632-1 CGM Standard. ○ Symbols for Emergency Management and First Responder communities – a map symbology set under development by the FGDC Homeland Security Work Group (HSWG). See: http://www.fgdc.gov/publications/homeland.html and http://www.fgdc.gov/HSWG/ ○ Geospatial Symbols (GeoSym) for Digital Displays. The map symbology set defined by NIMA to portray Vector Product Format (VPF) data. See: http://www.nima.mil/cda/article/0,2311,3104_12137_118865,00.html and http://www.nima.mil/ast/fm/acq/mil89045.pdf ○ MIL-STD-2525B is a Department of Defense Interface Standard that defines Common Warfighting Symbology http://symbology.disa.mil/symbol/mil-std.html.
Query Languages	
OGC Filter	○ OGC Filter Encoding Implementation Specification 1.0.0. OGC Document 02-059. ○ OGC WFS 1.0 and Filter 1.0 Change Requests. OGC Document 02-063.
Styling Description Languages	
SLD	○ SLD – Styled Layer Descriptor Implementation Specification 1.0. OGC Document 02-070. ○ SMS[3] – Style Management Services Discussion Paper. OGC Document 03-031.
SVG	○ Scalable Vector Graphic. W3C Recommendation http://www.w3.org/TR/SVG11/
Data Access Services	.

[3] The OWS1.2 Testbed initiative produced and tested significant enhancements to the SLD 1.0 specification and schema that are important to the technical approach in this initiative. Review the referenced SMS Discussion Paper (OGC Project Document 03-031) for more information about the enhancements to the SLD specification.

WFS	○ OpenGIS® Web Feature Service Implementation Specification 1.0.0. OGC Document 02-058. ○ OGC WFS 1.0 and Filter 1.0 Change Requests. OGC Document 02-063.
WOS	○ Web Object Service Discussion Paper, OpenGIS® Project Document 03-013.
Portrayal Services	
WMS	○ *OpenGIS® Web Map Server version 1.2* ○ OpenGIS® Map Context Documents Implementation Specification, version 1.0.
CPS	○ Coverage Portrayal Service (CPS) Interoperability Program Report. OpenGIS Project Document 02-019r1.
Catalog-Registry Services	
WRS./CS-W[4]	○ OpenGIS® Catalog Service Implementation Specification, version 1.1.1 ○ OpenGIS® Web Registry Service Interoperability Program Report. OpenGIS® Project Document 03-024.

[4] The Catalog Services Revision Work Group of the OGC Technical Committee is developing but has not yet released a new revision of the Catalog Services Implementation Specification 2.0 that will incorporate, as a profile, a “stateless” Web interface called CS-W that is derived from earlier work on WRS including implementations developed and demonstrated in previous Interoperability Program initiatives. So in addition to the Catalog Service 1.1.1 Implementation Specification, the WRS Interoperability Program Report (Project Document 03-024) will comprise the technical baseline for Catalog/Registry services for this initiative.

Appendix A: EMS-1 Architecture References

Refer to the OGC website (http://www.opengis.org/specs/?page=baseline) for the authoritative listing of adopted documents.

Note: Please contact the OGC Tech Desk if you need assistance in gaining access to these documents (techdesk@opengis.org).

OGC Specifications and Supporting Documents Relevant to EMS-1:

1. Style Management Services (SMS) Discussion Paper, OpenGIS® Project Document 03-031. Available at: http://www.opengis.org/specs/?page=discussion
2. Style and Symbol Management Services Requirements Interoperability Program Report, OpenGIS® Project Document 03-030. Available at: ***http://member.opengis.org/tc/archive/arch03.htm***
3. OpenGIS® Style Layered Description (SLD) Implementation Specification, version 1.0, available at: ***http://www.opengis.org/specs/?page=specs***
4. OpenGIS® Geography Markup Language (GML) Implementation Specification (version 3.0), available at: http://www.opengis.org/specs/?page=specs
5. OpenGIS® Web Feature Server (WFS) Implementation Specification, version 1.0, available at: ***http://www.opengis.org/specs/?page=specs***
6. OpenGIS® Filter Encoding Implementation Specification, version 1.0, available at: ***http://www.opengis.org/specs/?page=specs***
7. OpenGIS® Web Map Service (WMS) Implementation Specification, version 1.2, available at: ***http://www.opengis.org/specs/?page=specs***
8. OpenGIS® Map Context Documents Implementation Specification, version 1.0, available at: ***http://www.opengis.org/specs/?page=specs***
9. Coverage Portrayal Service Specification (CPS) Interoperability Program Report. OpenGIS Project Document 02-019r1, available at: ***http://member.opengis.org/tc/archive/arch02.htm***
10. OpenGIS® Catalog Service Implementation Specification, version 1.1.1, available at: ***http://www.opengis.org/specs/?page=specs***
11. OpenGIS® Web Registry Service Interoperability Program Report. OpenGIS® Project Document 03-024**.** Available at: ***http://member.opengis.org/tc/archive/arch03.htm***
12. Registry Service Requirements Interoperability Program Report. OpenGIS® Project Document 03-027. Available at: ***http://member.opengis.org/tc/archive/arch03.htm***
13. Web Object Service Discussion Paper, OpenGIS® Project Document 03-013. Available at http://www.opengis.org/specs/?page=discussion
14. Critical Infrastructure Collaborative Environment (CICE) Architecture: Enterprise Viewpoint Discussion Paper, OpenGIS Project Document 03-061. Available at http://www.opengis.org/specs/?page=discussion
15. Critical Infrastructure Collaborative Environment (CICE) Architecture: Information Viewpoint Discussion Paper, OpenGIS Project Document 03-062r1. Available at http://www.opengis.org/specs/?page=discussion
16. Critical Infrastructure Collaborative Environment (CICE) Architecture: Computation Viewpoint Discussion Paper, OpenGIS Project Document 03-063r1. Available at http://www.opengis.org/specs/?page=discussion

17. Critical Infrastructure Collaborative Environment (CICE) Architecture: Engineering Viewpoint Discussion Paper, OpenGIS Project Document 03-055r1. Available at http://www.opengis.org/specs/?page=discussion

Other OGC Specifications and Supporting Documents

18. OpenGIS® Abstract Specification, Topics 1-17), available at: http://www.opengis.org/specs/?page=abstract

19. OpenGIS® Reference Model, version 0.1.2, available at: http://www.opengis.org/specs/?page=orm

20. OpenGIS® Web Service Architecture Discussion Paper. OpenGIS Project Document 03-025, available at: http://www.opengis.org/specs/?page=discussion

21. OpenGIS® Grid Coverages Implementation Specification, version 1.0, available at: http://www.opengis.org/specs/?page=specs

22. OpenGIS® Coordinate Transformation Services Implementation Specification, version 1.0, available at: http://www.opengis.org/specs/?page=specs

23. Web Coverage Service Implementation Specification, Version 0.0. OpenGIS® Project Document 02-076r3, available at http://www.opengis.org/specs/?page=specs

24. OpenGIS® Location Services (OpenLS): Core Services [Parts 1-5], OpenGIS® Project Document 03-006r1. Available at: http://www.opengis.org/specs/?page=specs

25. OpenGIS® Location Services (OpenLS): Navigation Service [Part 6], OpenGIS® Project Document 03-007r1. Available at: http://www.opengis.org/specs/?page=specs

26. Service Information Model Discussion Paper, OpenGIS® Project Document 03-026. Available at: http://www.opengis.org/specs/?page=discussion

27. Web Terrain Service Discussion Paper, OpenGIS® Project Document 01-061. Available at: http://www.opengis.org/specs/?page=discussion

28. XML for Imagery and Map Annotations (XIMA) Discussion Paper, OpenGIS® Project Document 01-019. Available at: http://www.opengis.org/specs/?page=discussion

29. Web Coordinate Transformation Service Discussion Paper, OpenGIS® Project Document 02-061r1. Available at http://www.opengis.org/specs/?page=discussion

30. Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations, OGC Recommendation Paper, version 1.1.0, available at: http://www.opengis.org/specs/?page=recommendation

ISO Specifications

31. ISO 19101:2002 (Reference Model): http://webstore.ansi.org/ansidocstore/product.asp?sku=ISO+19101:2002

32. ISO 19107 (Spatial Schema) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19107_(E).pdf

33. ISO 19108 (Temporal Schema) : http://www.isotc211.org/protdoc/DIS/DIS19108.pdf

34. ISO 19109 (Rules for Application Schema) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19109_(E).pdf

35. ISO 19110 (Methodology for Feature Cataloguing) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19110_(E).pdf

36. ISO 19111 (Spatial Referencing by Coordinates) : http://www.isotc211.org/protdoc/DIS/DIS19111.pdf

37. ISO 19112 (Spatial Referencing by Geographic Identifiers) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19112_(E).pdf

38. ISO 19115 (Metadata) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19115_(E).pdf

39. ISO 19117 (Portrayal) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19117_(E).pdf

40. ISO 19119 (Services) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19119_(E).pdf

41. ISO 19123 (Schema for Coverage Geometry and Functions): http://www.isotc211.org/protdoc/211n1227/readme.htm

42. ISO 19125-1 (Simple Features Access - Part 1: Common Architecture): http://www.isotc211.org/protdoc/DIS/DIS19125-1.pdf

43. ISO 19125-2 (Simple Features Access - Part 2: SQL option): http://www.isotc211.org/protdoc/DIS/DIS19125-2.pdf

44. ISO 19128 (Web Mapping): http://www.isotc211.org/protdoc/211n1331/211n1331.pdf

Other Related Specifications:

45. Uniform Resource Identifiers (URI): Generic Syntax (RFC 2396) T. Berners-Lee, R. Fielding, L. Masinter, available at: http://www.ietf.org/rfc/rfc2396.txt

46. Extensible Markup Language (XML) 1.0, Second Edition, Tim Bray et al., eds., W3C, 6 October 2000. See http://www.w3.org/TR/2000/REC-xml-20001006

47. XML Schema Part 1: Structures. World Wide Web Consortium (W3C).. W3C Recommendation (2 May 2001). Available [online]: http://www.w3.org/TR/xmlschema-1/

48. XML Linking Language (XLink) Version 1.0, DeRose, S., Maler, E., Orchard, D., available at http://www.w3.org/TR/xlink/

49. Web Services Description Language (WSDL) Version 1.2. W3C Working Draft (9 July 2002). World Wide Web Consortium (W3C). Available [online]: http://www.w3.org/TR/wsdl12/

50. Simple Object Access Protocol (SOAP) 1.1, Box, D., et. al., available at http://www.w3.org/TR/SOAP/

51. UDDI – Universal Description, Discovery, and Integration, see http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec

52. Registry Information Model v2.1, OASIS/ebXML Registry Technical Committee (Approved Committee Specification, June 2002). See http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep

53. Registry Services Specification v2.1. OASIS/ebXML Registry Technical Committee (Approved Committee Specification, June 2002). See http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep