



Model for the Future of the e-GMS

Version 1.0, March 2005

Purpose of this document

- 1 This document has been produced by the Cabinet Office e-Government Unit's Metadata Technical Working Group (MTWG) to provide a formal¹ statement of its recommendations for the future development of e-GMS². The Model has been approved by the Metadata Working Group. Comments on the Model should be sent to govtalk@cabinet-office.x.gsi.gov.uk.
- 2 It describes a model for the development of the e-GMS. The model is intended to provide a framework within which the following requirements can be met:
 - Clear expression of the scope and usage of the e-GMS;
 - Clarification of the internal consistency of the e-GMS;
 - Clarification of the e-GMS's relationship with other standards (notably Dublin Core);
 - Continuity with e-GMS 3.0 (to the extent that this is coherent with the above).
- 3 The intention is that this model will be used for versions of e-GMS from version 4 onwards. This will require further detailed guidance, which will be developed by the MTWG.

Underlying principles

- 4 The model is based on the following principles:

Owners of information resources need to support certain functions with their resources; in order to perform these functions well, they need to maintain, use and make available appropriate metadata for those resources.

The e-GMS will not specify or constrain the way in which metadata is stored. It will specify only the way in which metadata must be presented or exposed in the context of specified functions.

- 5 The second principle has far-reaching consequences, all of which will contribute to making the e-GMS more widely applicable and easier to use than it is at present.
- 6 The principles – and the model described below – are consistent with the requirements for the e-GMS set out in the e-Government Interoperability Framework, which mandate that the e-GMS must be simple to use, compliant with international standards, extensible, inclusive, and capable of meeting the information retrieval and management needs of government (see e-GIF section 3).

¹ Note that “formal” in this document refers to careful and fit-for-purpose use of natural language rather than mathematical formality.

² This document is not itself part of the e-GMS and is written for e-GMS designers not e-GMS users.

The Model

- 7 The basic e-GMS model is outlined in Figure 1. In this diagram, the shaded box indicates the scope of the e-GMS; each block represents a distinct kind of information; and the arrows represent flows of information at design time. For example, the design of an Application Profile will draw on one or more Function descriptions, and will also use the Metadata Term Declarations. The boxes are described individually below.

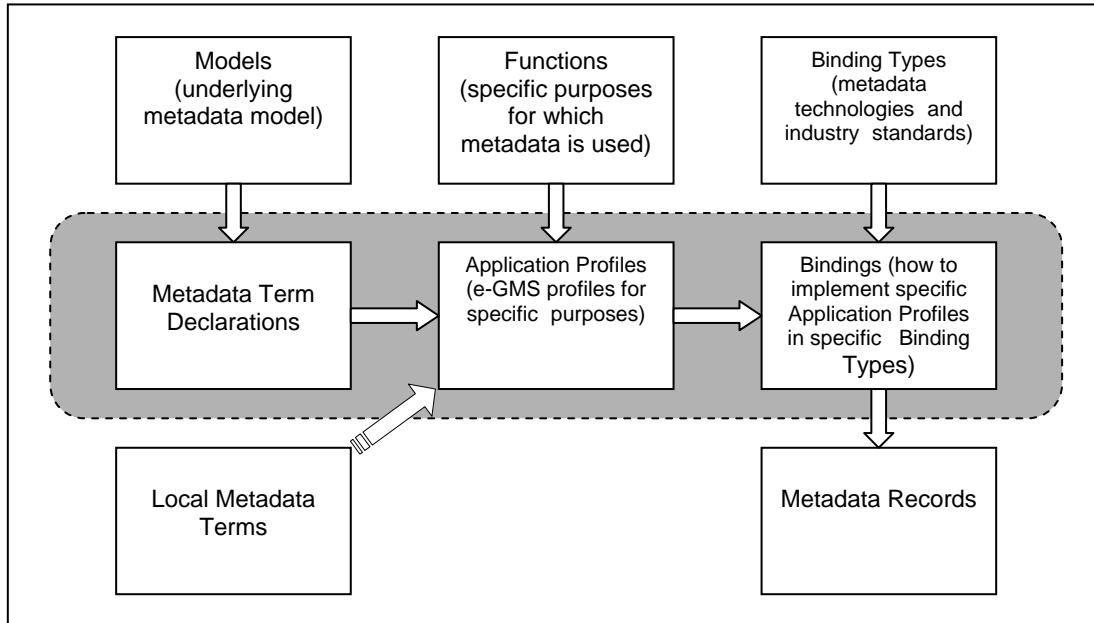


Figure 1: Basic e-GMS Model

Functions

- 8 “Functions” are significant activities performed with the resources (note that this usage differs from the conventional usage as in “business function”). Examples of these functions are likely to include “Web resource discovery” and “Transfer of records to The National Archives”. There will be others.
- 9 Each function will be defined formally, in natural language. The development of these functional definitions is a significant addition to e-GMS, as it currently contains no equivalent.³
- 10 There is no agreed list of functions at present. The intention is that a small number of key functions will be identified early on, as a basis for the development of e-GMS v4. At later stages, it will be possible to add more. Clearly, it will be important to ensure that the functions remain at a high level, are reasonably concise, and do not proliferate into an unmanageably large set.

e-GMS

- 11 The e-GMS is represented by the shaded rectangle. At present (i.e. in versions 3 and below), it describes metadata elements in general terms, with selected examples. For future versions, it will explicitly recognise “application profiles” and “bindings” of metadata. These are explained below.

Application Profiles

- 12 Certain metadata is required to perform functions. In practice, the metadata required to perform a function on a resource will be a subset of the entire metadata for that resource. In reality, the subset can be determined by other factors too, but

³ Some precise functional definitions that are similar to what is intended here do exist, for example the extensive functional description of electronic records management published by the National Archives.

for these purposes the simplification is made that the subset is determined by the function only. Each such subset is called an “application profile”. The application profile will be defined to meet the function’s requirements, including functional and standards compliance requirements.

- 13 There will be several metadata application profiles for a resource, so a one-to-many relationship between “resources and their metadata” and “application profiles”. They will overlap – for example, the element “Title” is likely to appear in all or most application profiles. Clarification of the content of each application profile will allow the different e-GMS user communities to take into account their specific needs without needing to be concerned about other communities’ metadata.
- 14 Some application profiles may apply to more than one function.
- 15 Application profiles will be defined in human-readable form (possibly in a form similar to the element definitions in e-GMS v3). However, an application profile will not be implementable without a “binding” – see Binding Types below.
- 16 The application profiles will be specific to one underlying model.

Binding Types

- 17 There are several ways in which metadata can be encoded and represented. For example, e-GMS v3 mentions Dublin Core embedded into HTML meta tags. Other possible encodings include XML and RDF/XML (the Resource Description Framework). Each of these is termed a “binding type”. A “binding type” specifies the syntax used to encode the metadata.
- 18 Ideally, the binding types approved and recommended for e-GMS will be specified in future editions of the e-GIF Technical Standards Catalogue.
- 19 Some application profiles can be represented usefully with more than one binding type; and most binding types will be applicable to several application profiles. The resulting Binding specifications are represented in the model by the “binding” block.

Bindings

- 20 A binding is an application profile, as defined above, represented according to the specifications of one binding type.
- 21 This combination of application profiles and binding types will mean that software developers will be able to build software that complies with a wide range of metadata standards, some of which are conceptually incompatible, while retaining e-GMS compliance.
- 22 Bindings will be defined in human-readable form, though they inevitably will require their readership to have a deeper technical knowledge than will other parts of the e-GMS. Supplementary specifications such as XML schemas will be provided where required.

Metadata Records

- 23 This represents the metadata records associated with the resources held by an organisation. e-GMS does not define how resources or metadata are stored. However, interoperable interfaces to these resources require the exposure of records that conform to specific e-GMS bindings.

Models

- 24 A metadata model defines the underlying structure of metadata records used within a particular business, and the relationship of the constituent parts of a description to the resource(s) being described. Examples of models include the Dublin Core Abstract Model and the IEEE LOM Data Model.

- 25 The model determines the metadata terms required to make up a valid description. For example, the Dublin Core Abstract Model defines the use of "elements", "element refinements", "encoding schemes", etc.

Metadata Term Declarations

- 26 The terms used to construct metadata descriptions within a particular model must be declared before they can be used. Such declarations can take the form of human-readable documents or machine-readable 'schemas' or both.
- 27 Typically, a core set of terms is made available by a global standards body for widespread use (e.g. the DCMI or IEEE) with additional terms being declared separately by particular national or local initiatives.

Local Metadata Terms

- 28 Most organisations using the e-GMS will need to add business specific, or "local" metadata terms. The e-GMS will allow for such additions. It will be up to users organisations to manage their addition and maintenance.

Next steps in development of the Model

- 29 The next step will be for the Metadata Technical Working Group to develop:
- A standard format for the definitions of functions
 - The rules to be used in defining application profiles, and the way in which application profiles are presented (note the CEN Workshop Agreement in this area).
 - Rules defining how application profiles may be extended without breaking interoperability.
 - The rules to be used in defining bindings, and the way in which bindings are presented.
 - An approach for the publication of the e-GMS using this model.
- 30 Simultaneously (or as soon as possible), an appropriate authority will need to:
- Define the first few functions to be specified fully.
- 31 Once these initial steps have been completed, the next step will be to start to convert e-GMS v3 to a new format consistent with this model. This work will be done by the appropriate group, presumably the eGU Metadata Working Group supported by the MTWG.

Implications for the e-GMS

- 32 This model calls for a modular e-GMS. Aside from introductory text (which will need extensive expansion), it eventually will consist of several application profiles and bindings.
- 33 The ability to define new application profiles and bindings will need to be controlled. The governance regime for this has yet to be defined.

Other issues

- 34 When specifying requirements for system procurement, it will no longer be sufficient to require merely "e-GMS compliance." Instead, specifiers will have to require compliance to some selected application profile/binding combination(s).