# Semantic Frameworks for e-Government

Charles Crichton, Jim Davies, Jeremy Gibbons, Steve Harris, and Aadya Shukla
Software Engineering Programme, University of Oxford
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
Jim.Davies@comlab.ox.ac.uk

## ABSTRACT

This paper explains how semantic frameworks can be used to support successful e-Government initiatives by connecting system design to a shared understanding of interactions and processes. It shows how metadata standards and repositories can be used to establish and maintain such an understanding, and how they can be used in the automatic generation and instantiation of components and services. It includes an account of a successful implementation at an international level, and a brief review of related approaches.

## Categories and Subject Descriptors

J.1 [**Administrative Data Processing**]: Government; H.3.5 [**Online Information Systems**]: Web-Based Services; J.3 [**Life and Medical Sciences**]: Medical Information Systems

## General Terms

semantic web, ontology, metadata, model-driven

## 1. INTRODUCTION

Successful e-Government initiatives are built upon an adequate understanding of the interactions and processes that are to be supported. This understanding needs to be *shared* by a community of stakeholders, users, and developers from a range of organisations and disciplines; it needs to *evolve* to incorporate new knowledge, as well as changes in policy and structure; and it needs to be *recorded* and made available in the form of accurate, up-to-date documentation.

The basis of a shared understanding is a common semantics: the community needs to agree, to some extent, upon the meaning of key terms or actions. Natural language is sufficient for such a semantics only when the concepts are straightforward, the community is small or homogeneous, and the period of time over which understanding must be maintained is short. For complex problems, large or diverse communities, or long-lasting collaborations, a more formal approach is required.

At a bare minimum, we require something equivalent to a dictionary, providing standard definitions for the community to use. A definition might explain the meaning of a word, the purpose of a piece of data, or the intended interpretation of a message or action. An informal explanation may be supplemented by more precise statements: categorising or classifying the elements being defined; formalising relationships between them; and explaining how they might be used in combination.

e-Government requires a significant degree of formalisation and *computerisation* of semantics. The size of the community, the rate of evolution, and the importance of documentation make it essential that the semantics can be accessed, maintained, and incorporated into delivered systems without the need for extensive, error-prone manual intervention. This paper explains how this can be achieved, at no additional cost, through the use of simple, practical frameworks for semantics-driven development,

A practical, semantic framework can be defined in terms of constructs at three different levels: *terminology services*, *metadata registries*, and *model repositories*. The first level presents a collection of defined terms, structured in a way that suits one or more possible applications. For example, a terminology for education might include terms such as *institution* and *qualification*, record that the terms *university* and *high school* denote particular kinds of institution, and record also that the terms *master's degree* and *international baccalaureate* are related in some way to the notion of institution.

The second level presents a collection of 'metadata elements', each of which describes a measurement or observation. A metadata registry for education might include elements such as *institution attended*, *full title of degree awarded*, and *result obtained*. Each element may be related to one or more terms in the underlying terminology, and additional semantic information is provided by informal explanations of intended purpose and an association with a domain of possible values. The registry also records relationships between elements, such as equivalence, specialisation, and versioning.

The third level presents re-usable models for the definition of information artifacts, such as database schemas, service descriptions, forms, queries, and reports. A model repository for education might include models of admissions forms, study transcripts, and spreadsheets for reporting registration and progress data to national agencies. The fields on


*ICEGOV* 10–13 December 2007, Macau.

the forms, the entries on the transcripts, and the columns on the spreadsheets may be described, and given computable semantics, by linking them to the data elements in a metadata registry.

In this paper, we clarify the interpretation and application of semantic frameworks. In particular, we discuss the extent to which terminologies need to be developed and presented as *ontologies*, balancing the benefits of additional semantics against the costs of further specialisation. We explain the information content of metadata elements, and the role of a default model for a metadata registry. We explain also the way in which models and ontologies for a particular domain can make effective use of terminology and semantic metadata.

We show how practical, semantic frameworks can be implemented using a combination of open source technologies, and how the contents of metadata registries and model repositories can be developed and maintained using standard office tools. We show how the resulting models and metadata can be used in the development of information systems that accurately reflect a shared understanding, and can be efficiently updated to take account of changes in requirements. We end with a discussion of related work: in particular, that related to metadata standards.

## 2. BACKGROUND

Increasing reliance upon electronic communication, together with the ambitions and demands of a global information society, means that *e-Government* is becoming the expected means of implementation for government policies, activities, and initiatives. Although considerable progress has been made, the reputation of public sector information technology remains poor. Most people can quote at least one high-profile disaster, in which a large e-Government project singularly failed to deliver.

Examples include: US Government's *Talon* threat reporting system, established in 2002 and closed in 2007 [4]; the *Pathway* benefits card system, abandoned with losses of £1 bn [20] and severe consequences for the UK post office; and the UK Child Support Agency system, which led to the collapse of the agency it was intended to support [15].

The challenges in developing information technology for public sector applications are no different from those encountered in other large, enterprise computing initiatives. They are, however, exacerbated by three main factors: the likelihood of conflict and misunderstanding between different stakeholder groups; the fact that requirements are linked to changes in policy and legislation; and the expectation that data and processes should be accessible, and also compatible with those in other initiatives.

These problems have been addressed in part through the wider use of XML [14] and associated technologies. For example, the use of agreed XML tags, rather than specific characters in fontsets, greatly improves data accessibility and interoperability in multi-lingual contexts, particularly if the fontsets themselves are proprietary [24].

However, while XML can provide a common means of representation, there is still considerable scope for confusion as to the meaning or *semantics* of both tags and schemas. The use of namespaces provides a very basic level of semantics, by identifying the domain of discourse, but different people within a domain may have very different ideas as to intended function or interpretation of the same piece of data.

Some initiatives, such as the UK electronic Government Interoperability Format [29], give an informal semantics for common terms by explaining the intended meaning of a collection of XML tags in a series of publications. This is useful documentation, but falls short of our expectations in terms of flexibility and accessibility: a structured, computable representation is essential if we wish to adopt and maintain rich terminologies across multiple initiatives.

To properly present the semantics of an element of data, we need to consider the components and processes that make use of it. Conflicts and misunderstandings can be resolved, or at least identified at an earlier stage, if aspects of structure, functionality, and interpretation are conveyed through the use of models. This is standard practice in software engineering, although the audience for the model is usually quite restricted, and thus much of the detail, or semantics, may be left to verbal communication.

In e-Government initiatives, models not only reduce the potential for misunderstanding, but also reduce the cost of adapting the system to address changes in requirements: they provide a clear description of the relationships between elements of the system, above the level of the executable code; they also suggest re-usable components or aspects of the design. They may also serve as a record of design intentions, addressing a range of governance concerns, from accountability to accurate, accessible documentation.

Experience has shown that a common terminology, and the use of models, is not enough to provide for the degree of data interoperability and re-use that is expected of e-Government systems. What is required is the adoption of a complete, semantic framework linking terminologies and models via an intermediate layer of metadata elements: templates for elements of data, measurements, or observations with agreed, computable semantics. These elements are explained partially in terms of an agreed terminology, and give a formal meaning to the data attributes in our models.

In this area, e-Government initiatives can usefully benefit from work undertaken in the scientific community — most notably, the data sharing initiatives in fields such as cancer research [3]. In many cases, key scientific questions cannot be answered unless data from multiple centres and studies can be integrated and analysed; interoperability is not merely a worthy goal, but now an essential requirement for progress. The approach advocated in this paper has been employed in the development of an international infrastructure for cancer research informatics.

## 3. SEMANTIC FRAMEWORKS

For our purposes, a semantic framework consists of components at three levels: terminology services, providing interpretations for basic terms; metadata registries, holding collections of observations; and model repositories, descriptions of components or data sets, or characterisations of domain information.

### 3.1 Terminologies

A *term* is a word or phrase used in a definite or precise sense in a specific subject [22]. To use a term consistently, we require some kind of definition: at the very least, we need an indication of the context in which it is being used. For

example, we might label the term `paper` with a particular identifier to indicate that it is being used in the context of an academic conference, and with a different identifier to indicate that it is being used as a type of material.

The word *terminology*, in its usual sense [22], means nothing more than a collection of terms intended for a specific purpose. However, as terms are of little use without definitions, we should expect to see this word used interchangeably with either *glossary* or *dictionary*, words that suggest a collection of terms with corresponding definitions. We will use the word *concept* to denote the combination of a term and a definition, perhaps with additional, qualifying information.

Concept modelling tools such as Protégé represent collections of concepts not as simple glossaries, but as *ontologies*, in which relationships are recorded between concepts, expressing notions such as 'broader than', and 'narrower than'. These relationships can be used to extend the definitions of terms, which has the effect of further constraining their interpretation. Clearly, any further constraint or specialisation may reduce the applicability of the terminology, and could easily introduce contradictions.

For example, we might decide that things labelled `paper` may be categorised into two disjoint subclasses—journal papers and conference papers—only to find later that a conference paper may be included when the conference proceedings are published as a special issue of a journal. If our categorisation is for the purposes of reporting publication impact, then we might find that the paper should be counted as a journal publication; if, on the other hand, it is for the purposes of recording publication activity, and perhaps related to travel, then it might be recorded as a conference paper. Either way, if we record the disjoint relationship as part of our terminology, then that terminology can be used for only one of the two purposes.

For this reason, we advocate the inclusion of only those relationships that are necessary for the *organisation* and *maintenance* of the terminology. For the former purpose, we suggest the use of the 'broader than' and 'narrower than' relations, together with a some notion of being 'related', and a record of alternative terms or synonyms. This is exactly what is provided by the Simple Knowledge Organisation System (SKOS) [32] core vocabulary. Although some semantic information is presented here, it is quite limited in extent, and should not unduly constrain the applicability of the terminology.

If we expect the terminology to be used to support development activity and organisational processes across a large, heterogeneous community, such as we would expect to encounter in e-Government, then we must provide effective mechanisms for updating it in the face of changes in requirements or understanding. In most cases, the only acceptable form of update is extension: once a term has been used, we may need to access the definition corresponding to that usage at any time in the future. Our terminology system should allow us to maintain multiple versions of terms, definitions, or concepts. It may also record relationships between these versions: in particular, indicating when one definition should supercede another.

The practical concern of maintenance—the fact that we expect our terminology to be used, and to be updated regularly to support that usage—mitigates against the inclusion of additional, semantic information as relationships between concepts. When we update a concept, we must consider also any relationships that this concept appears in: in the extreme case, we might need to consider every other concept in the terminology. Making our terminology into a rich, domain-specific ontology not only reduces its applicability, it also makes it more difficult to manage and maintain.

Even the relation of 'narrower than' can be problematic, when given semantic importance (rather than regarded simply as means of organising concepts). For example, the SNOMED-CT clinical terminology has multiple inconsistencies in the use of `is_a` between the classes used to represent medical concepts. The likelihood of such inconsistencies means that the recording of a relationship may be of little semantic value: we cannot use it, reliably, to support reasoning about data that is collected.

As a simple example of what may appear in a terminology intended to support an e-Government initiative, consider the following definition of a postal code, based upon that used by the Canadian Radio-television and Telecommunications Commission [9]:

```
term: postal code
definition: codes used by postal services
  to divide large geographic areas into
  discrete zones in order to simplify
  delivery. (...) French: code postal.
```

In a terminology service such as LexBIG [18], this would be packaged and delivered, together with namespace and identifier attributes, and provenance and maintenance information, as an object of class `Concept`.

## 3.2 Metadata elements

An agreed terminology provides a sound basis for communication and description, but the information that we record is unlikely to consist of usages of terms. Instead, it is acquired and processed as elements of data: well-defined records of specific observations or measurements. To build a shared understanding, we must agree not only upon terminology, but also on the observations made using that terminology. The second component of our proposed semantic framework is a registry of *metadata elements*: re-usable templates for observation or measurement.

The potential benefits should be clear: if the same template is used for the acquisition of data in two situations, then we can be sure that the data obtained is compatible—these are two instances of the same observation. An international standard [16] exists for metadata registries, incorporating a similar notion of metadata elements, and several examples of registries have been constructed, most notably in the domain of healthcare informatics: by the US National Cancer Institute [21], the Canadian Institute for Health Information [8], and the Australian Institute of Health and Welfare [2].

For example, a common observation made of patients under care is the *World Health Organisation (WHO) Performance Status*. This is a measurement of the extent to which a patient is able to perform everyday activities, expressed as an integer value between 0 and 4; criteria are supplied to support the evaluation, with 0 corresponding to 'fully active, able to carry on all pre-disease performance without restriction' and 4 corresponding to 'confined to bed or equivalent'. The standardised name, value domain, and criteria mean that we can rely on the same observation being made in dif-

ferent circumstances, and thus safely assume that the data collected has consistent semantics.

Clearly, if a metadata element is to be re-used effectively, then it must be easy for potential users to locate it within a repository and to assess its suitability for purpose. We will require some means of organising the elements in a repository, usually in one or more hierarchies, focussed upon patterns or areas of expected usage (rather than any intrinsic semantics). We will require also support for maintenance and evolution: as with terminologies, we would not expect to delete a metadata element from a registry, merely to add a new element that supercedes it; we would need to record relationships between elements to support version control.

However, it is not enough to simply record that one observation supercedes another, or is related to it in some way: in realistic applications, we will need to record whether or not two different observations are compatible, and the ways in which data recorded in one observation may be translated to fit the specification of a compatible observation. A simple example is provided by two alternatives to the *WHO Performance Status*:

- the *ECOG* performance status has the same criteria, but adds an additional value 5, indicating that the patient is dead;
- the *Karnovsky* score is a value between 0 and 100, with 0 indicating death, and 100 indicating 'perfect appearance of health'.

Our metadata registry may include a description of the relationship between these elements, specifying exactly how an ECOG status or a Karnovsky score can be translated into a WHO performance status: the present version of the ECOG status metadata element is shown as Figure 3, appended. In either case, the translation will itself be the subject of agreement, and possible revision, across the community.

As a simple example, consider the following metadata element associated with postal addresses in the UK:

```
name: UK post code
definition: <postal code> identifying a small
  number of co-located properties in the UK
value domain:
  datatype: STRING
  input mask:
    [A-Z]{1,2}[0-9R][0-9A-Z]? [0-9][A-Z-[CIKMOV]]{2}
```

Notice how the definition of the element includes a reference to an agreed piece of terminology. A related metadata element, with the same reference, might be

```
name: US zip code
definition: <postal code> identifying a small
  number of co-located properties in the US
value domain:
  datatype: STRING
  input mask: [0-9]{5} (\-[0-9]{4})?
```

The ISO 11179 standard for metadata elements and registries includes provision for a default model or ontology: the above elements may be extended to refer to a specific property of a class, and the specified relationships between this and other classes may assist in the organisation of elements, and the maintenance of the registry.

As in the case of a terminology, it is unlikely that we would wish to regard this model as a definitive extension to the semantics provided by the definitions in the registry: it would be all too easy to over-constrain the application of what are intended to be *common* elements of metadata, simply by according them an over-specific meaning; it would also be easy to introduce inconsistencies.

## 3.3 Models

A practical, semantic framework will include a repository of object and class models. These can be used to describe components, interfaces, and other artifacts associated with the design and implementation of systems, and to add further semantics to metadata elements for application within a specific domain. For example, a model repository may include a model of a form used to collect data for a particular purpose:

```
CLASS UKPassportApplication
  ...
  applicantAddressPostCode : <UK post code>
  ...
```

By using a metadata element from the registry, we are ensuring that this aspect of the model semantics will be consistent with any other model that uses a `UK post code`. The model repository provides another level of re-use: we may use the class `UKPassportApplication` as part of the specification of an on-line application service, but also as part of the specification of a system intended to support immigration records, or inter-government collaboration on security.

Alternatively, a model may be used to describe a particular domain of application for metadata elements, in which the usage of elements is further constrained, without needing to describe any particular design or implementation. For example, a model might record that the metadata element `performanceStatus` is a `patientObservation`. With this particular ontology, the element `performanceStatus` can be used only to refer to observations of people in clinical care, even though the same metadata element may be used to refer to observations in a wider context. The ontology may be ideal for characterising clinical data, or for use in the design of a clinical information system, but may not apply in a social care setting.

A model repository can support re-use at an even higher level through the inclusion of generic models or metamodels: templates that can be instantiated to produce parameterised models for specific purposes. For example, we might present a metamodel for a generic application service, ready to be instantiated with a combination of components or metadata elements, to produce services for on-line passport application, driving licence application, or for residency permits. This degree of re-use not only reduces the cost of systems development, but also increases the extent to which data can be automatically integrated.

Figure 1 shows the relationship between the three components of the semantic framework: attributes in the model repository are defined by reference to metadata elements; names in the model repository and metadata registry may be linked to definitions in the terminology service. Models and model fragments in the repository may be used to generate components of the information system, which may itself make use of registry and terminology services at run-time.

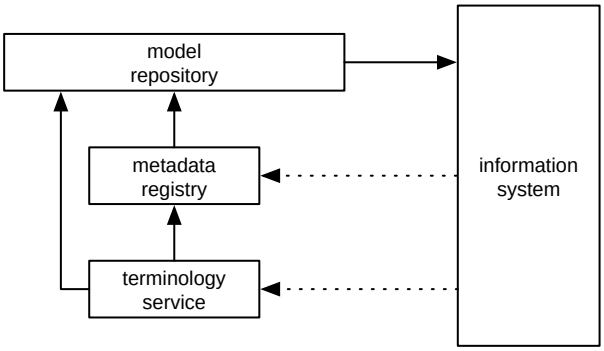


**Figure 1: A semantic framework**

# 4. SEMANTICS-DRIVEN ENGINEERING

The value of a semantic framework is increased considerably if models based on metadata elements and terminology are used to configure or generate components of information systems. We are then able to build systems in which the semantics of the data is to be accessible, computable, and—if we use common metadata and terminology—compatible by design. Alongside this, there are the expected benefits of model-based or model-driven software engineering [26], including a significant reduction in the cost of development and maintenance.

## 4.1 Model-based development

Most of the documents associated with the design of a system are not intended as formal, accurate models of structure or functionality. Instead, they are fragments of specification, written in natural language, and presented as reports, spreadsheets, and diagrams. These are partial descriptions, often containing apparent contradictions, and there is no prospect of using these to generate a system *automatically*.

Yet these are the documents that inform decisions such as those on whether to proceed, on project scope, on supplier selection, and on contract fulfilment, and it is here that a semantic framework can start to produce real benefit. Reports and spreadsheets in which key terms are annotated with a link to agreed terminology, and data elements are annotated with a link to detailed semantics in a metadata registry can be concise and unambiguous, while making explicit a shared understanding of exactly what is required.

In development, more formal models—typically, object models and service descriptions—can present precise descriptions of structure and functionality in which data attributes have an accessible, computable semantics, and terms have an agreed meaning. It may then be possible to determine programmatically—at the design stage, or after deployment—whether two systems are holding data that has exactly the same semantics: an essential prerequisite to the systems integration required for *joined-up government* [6], in which central and local government, different departments and agencies, work together to tackle social problems.

The means of accessing and incorporating semantic information is through the medium of *plug-ins* for the 'office' applications used in the production of text documents and spreadsheets, and for the modelling tools used in the development of information systems. The means of preserving it is through annotation of structured data, typically presented in XML: now the interchange language of both office systems and modelling tools. Figures 4 and 5 in the Appendix show how metadata elements can be used and incorporated in Microsoft Excel and in the Sparx Enterprise Architect modelling tool, respectively.

As an example, consider the situation of a proposed survey of water quality. The proposal document might include links to detailed descriptions of specific measurements: measures of pH, colour of water, taste and odour, heavy metals. A reviewer of the proposal might suggest the inclusion of additional or alternative measurements—of pesticides, or hormone analogues—to enable comparison with other results, or to make the data collected useful for another purpose. Once the survey was complete, anyone who wished to examine the data would have access to the meaning of the values recorded. In this way, the value of the survey may be both increased and assured.

An additional advantage of the model-based approach is that semantic information can be made available at runtime. This can be used, dynamically, to determine the behaviour of interfaces, services, or access control systems. It can also be used by those who are entering data: for example, if a user wishes to make a report, they can use the terminology services to help them choose phrases with appropriate, agreed semantics. For example, the terminology and metadata elements set out in the US *Global Justice XML Data Model* [30] might assist a user in reporting the type of a stolen vehicle in terms that can be immediately understood and processed by staff and information systems in other countries.

## 4.2 Technology

The approach described above has been validated through the development and application of semantic frameworks for clinical research. The UK *CancerGrid* [7] project has constructed a framework that exploits existing terminologies to support a metadata registry based upon an open-source XML database.

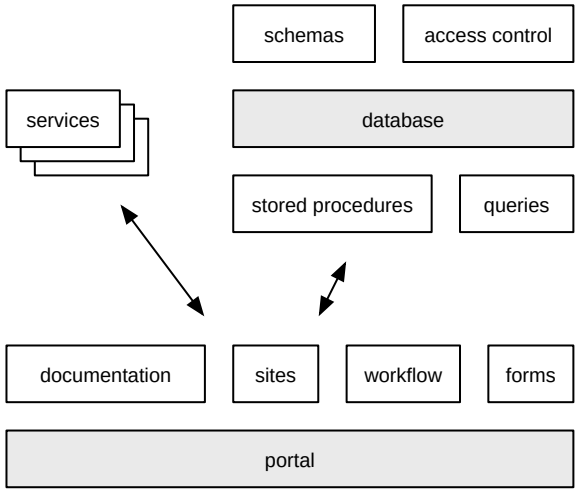


**Figure 2: Implementation architecture**

*Metamodels* for the specific domain—cancer clinical trials protocols—are constructed in UML using standard modelling tools, and instantiated, interactively, to produce mod-

els for specific experiments. The instantiation tool makes use of the metadata registry and terminology services: the user can choose specific observations and measurements from the registry; the same interface also supports extension of the terminology, and the registration of new measurements.

The instantiation tool (the 'protocol designer') can also make use of the contents of the model repository—in the shape of fragments of forms, documents, and workflow. The instantiated model is stored as XML, and used as the source for a collection of generators, producing implementations of services, and configuration files for databases and portals. The generators are written in the standard, open language of XSLT [28] (Extensible Stylesheet Language Transformations), and their action is coordinated using the framework language of the portal—usually Java or C#.

The architecture of the resulting implementation is shown in Figure 2. The shaded components—the database and the portal—are off the shelf; the white components, from the services to the documentation—are generated from the model. In the case of CancerGrid, the services generated include: patient registration, entry into a clinical trial, data collection, trial management monitoring, drug allocation, calendar management, and adverse event reporting.

Experience of modelling and generating systems suggests that it is impractical to incorporate every aspect of system design within a metamodel. The artifacts that are generated often require some manual customisation before final deployment: this is particularly true of interface issues, such as the arrangement of boxes on an information screen. The costs of encoding this information, and of entering it as part of the model, would not justify the benefits; a better approach is afforded by a generation technology that carefully avoids overwriting customised aspects of the generated artifacts — what Vlissides [31] calls the *generation gap* pattern.

The metamodelling approach used in CancerGrid offers an additional degree of flexibility in standardisation: rather than insist upon a single model for trials, we are able to require that trial models should conform to a generic template—the trials metamodel. This metamodel describes features and policies at the level of design principles: in the case of clinical trials, these principles are set out in an international standard called the CONSORT statement [19]. We might expect similar templates of good practice to be established for e-Government, and to find that the metamodelling approach is equally applicable.

An important consideration in the adoption of semantic frameworks and model-based approaches is the integration of generated technology with existing artifacts. Legacy tools and systems will produce and consume data without any semantic annotation. A practical solution to this problem is to create XML mappings from data attributes—described, for example, by XPath expressions—to the identifiers of elements in a metadata registry. This allows us to (re-)unite data with its semantics on the fly.

The full CancerGrid technology stack has been used in support of a large-scale cancer clinical trial in the UK, and aspects of the technology have been adopted by the Veterans Health Administration in the US. The terminology services and metadata registry have been used effectively alongside those of the US National Cancer Institute [3] in trials design and in adverse events reporting.

A complete implementation using open source technology is available, using Java and Grid technology for the portal. At least as important, in terms of potential adoption, is the fact that the technology stack will work equally well with Microsoft SharePoint [13] server as the portal technology, allowing organisations that have already licensed standard Microsoft technology to enrich their services and data with well-defined semantic information.

# 5. DISCUSSION

In this paper, we have set out a precise notion of semantic framework, and explained how it might be used to address some of the challenges inherent in e-Government initiatives. The idea of establishing and deploying a common, computable semantics for data is hardly new: consider, for example, the continuing interest in the 'semantic web' [5]. However, many of the existing efforts have achieved only limited adoption, due to

- over-complex terminologies and models, presented as rich ontologies: poorly understood, regarded as unsuitable, expensive and difficult to implement;
- lack of emphasis upon, or technological support for, the re-use of common metadata elements, resulting in large collections of unrelated measurements;
- lack of synchronisation between models and implementations, often with implementations 'hard-wired' to use outdated versions of models, metadata elements, and terminology.

Our three-level formalisation, in which

- any relationships expressing semantic constraints on application are described by ontologies at the model level, not presented as part of the shared terminology,
- metadata elements describe observations or measurements, and not terms or concepts (although their semantics may be explained in part through the use of shared terminology), and
- models, in which data is classified using metadata elements, are used as the basis for the configuration or generation of system components and interfaces,

is intended to address these concerns. The technology stack developed originally for the *CancerGrid* project has proved the effectiveness of this approach within the domain of clinical research informatics; we are now hoping to extend and refine the approach through wider adoption.

## 5.1 The role of XML

The requirement to support a diversity of cultures and languages, coupled with the need to access data from a variety of legacy information systems, has led to the widespread adoption of XML technologies in both industry and government [10, 27]: agreed markup provides for a common interchange format, replacing earlier electronic data interchange standards in the e-Government domain [25].

XML can serve as the implementation language for much of our semantic framework: it can be the representation format for concept descriptions in our terminology, for metadata elements in our registries, and for models in our repositories; it provides a means by which semantic information can be associated with data; and associated technologies can

be used in the generation of service descriptions, interfaces, forms, and documents.

A typical example of how XML can be used to record contextual or semantic information is provided by the EULEGIS system [17], which retrieves legal information from databases across Europe, written in different languages, and recorded in the context of different legal systems. The current focus is upon accessibility of documents, as in the Danish Infostructurebase [12] and the UK e-GIF framework [29]; however, there is an increasing emphasis upon semantic relationships and ontologies—something that we might see as misguided or premature without the structure provided by a semantic framework.

## 5.2 Domain modelling

A common problem encountered in the establishment of a semantic framework is that of domain modelling. The information requirements of e-Government range over every area of human endeavour, and are continually expanding, and it is thus impractical to produce a complete characterisation of information and meaning. Even public administration *theory* lacks agreed standards, definitions, and vocabularies [1].

Thus, rather than adopt the ontological approach of attempting to establish an understanding of knowledge within a domain [11], we focus upon a semantics of individual measurements and observations, extended and reviewed according to need: that is, through the design and development of specific applications and initiatives.

By positioning ontologies as models, we allow the metadata elements to be re-used regardless of the relationships that may hold between them in a specific domain. If the user wishes to specify that those relationships should hold, they can do so by accessing an appropriate ontology from the model repository. Models and ontologies can be tied to more specific purposes, and can evolve separately from the underlying metadata and terminology. More important, ontologies need not address the whole of the conceptual domain, only those aspects that are being used in the design of information systems.

## 5.3 ISO 11179

ISO 11179 is an international standard for metadata registries. It extends our summary of a metadata element with explicit notions of 'data element concept' and 'conceptual domain'. For example, our metadata element for a `UK post code` would be extended with

```
data element concept:
  object class: UK address
  property: postal code
conceptual domain: CONCEPTUAL DOMAIN NAME
```

Our recommended interpretation of these is as a means of organising the registry, rather than an opportunity to constrain the use and interpretation of metadata elements within a particular model or ontology.

There is no reason, for example, why the registry should impose a model for applications in which a `UK post code` is part of a `UK address`, which is a particular form of `Address`, rather than one in which a `UK post code` is a `Postal code`, which is part of an `Address`. That is, we might settle upon one hierarchy for the organisation of the registry, but allow users to choose another that is better suited to their particular domain of application.

A minor shortcoming in the current version of ISO 11179 is the fact that the same metadata element may be associated with values that have more than one meaning, when in fact a change in value interpretation or value range should produce a different metadata element. The standard is also difficult to translate into a specification of a compliant database implementation, due to the use of inheritance in the metamodel, where composition would be more appropriate. An additional complication is the lack of alignment with XML schema datatypes: the de facto standard for primitive type representation.

The registries constructed for the CancerGrid project, and the National Cancer Institute's caBIG initiative, are not the only implementations of ISO 11179. For example, the US National Information Exchange Model (NIEM) [23] is an e-Government initiative to improve communication across departments and agencies, with a particular focus upon emergency management, immigration, and intelligence.

The initiative defines 'data components' (metadata elements) to be reused across *information exchange packages*—models of messages expressed using UML and/or W3C XML schemas—passed between agencies. Each package is associated with an *information exchange package documentation (IEPD)* which describes the message in natural language, UML via an XMI file, and in tabular format. Tools are provided to support the creation and maintenance of NIEM standards, including a model browser, a mapping tool, and a schema builder.

The factorisation into terminology, metadata elements, and models sets the NIEM apart from other schema-based initiatives, such as the UK e-GIF framework, and has the clear potential to support semantically-reliable data exchange. It is more limited in its ambition than the frameworks developed for CancerGrid or caBIG, but is nevertheless entirely consistent with the vision that we set out here.

# 6. CONCLUSION

We have presented our approach to semantic frameworks for data-sharing communities. The approach is designed around: a common terminology, collecting agreed interpretations of shared vocabulary; a metadata registry, describing and relating terms in the vocabulary; and a model repository, providing fragments of system artifacts that exploit the common terminology and metadata. It is based on open standards, and is implementable in the form of plug-ins for off-the-shelf office tools. We have described how models assembled from artifacts in the repository may then be used to generate and configure components of information systems, supporting application integration by construction. The approach has been validated in the domain of clinical trials; we put it forward as a basis for e-Government initiatives.

# 7. REFERENCES


[1] G. Arango and R. Prieto-Diaz. *Domain Analysis: Acquisition of Reusable Information for Software Construction.* IEEE Computer Society Press, 1989.

[2] Australian Institute of Health and Welfare. Metadata online registry. `http://meteor.aihw.gov.au/`, Jan. 2007.

[3] H. Bal and J. Hujol. *Java for bioinformatics and biomedical applications*. Springer, 2007.

[4] BBC News. US to shut anti-terror database. `http://news.bbc.co.uk/1/hi/world/americas/6958791.stm`, August 2007.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):28–37, 2001.

[6] V. Bogdanor. *Joined-up Government*. Oxford University Press, 2005.

[7] R. Calinescu, S. Harris, J. Gibbons, J. Davies, I. Toujilov, and S. B. Nagl. Model-driven architecture for cancer research. In *Proceedings of the 5th IEEE International Conference on Software Engineering and Formal Methods (SEFM 2007)*, 2007.

[8] Canadian Institute for Health Information. Data dictionary. `http://secure.cihi.ca/ddexternal/`.

[9] Canadian Radio-Television and Telecommunications Commission. Telecommunications industry data collection glossary. `http://www.crtc.gc.ca/dcs/eng/glossary.htm`, Apr. 2007.

[10] J. Carmel. Drafting legislation using XML at the US House of Representatives. In *Proceedings of the International Conference on XML*. IDEAlliance, 2002.

[11] B. Chandrashekharan, J. Johnson, and V. R. Benjamin. What are ontologies, and why do we need them? *Intelligent Systems and Their Applications*, 14, 1999.

[12] Danish e-Government Project. Danish InfoStructureBase. `http://isb.oio.dk/info/`, 2005.

[13] B. English. *Microsoft Office Sharepoint Server 2007: administrator's companion*. Microsoft Press, 2007.

[14] E. R. Harold and S. W. Means. *XML in a Nutshell*. O'Reilly, 2004.

[15] House of Commons Committee of Public Accounts. Child Support Agency: Implementation of the Child Support Reforms. Technical Report 37th Report, The Stationery Office, July 2007. `http://www.publications.parliament.uk/pa/cm200607/cmselect/cmpubacc/812/81202.htm`.

[16] International Organization for Standardization (ISO). ISO 11179. Information technology – Specification and standardization of data elements. `http://www.iso.org/`.

[17] V. Lyytikainen, P. Tiitinen, and A. Salminen. Challenges for European legal information retrieval. In *Proceedings of the IFIP 8.5 Working Conference on Advances in Electronic Government*. University of Zaragoza, 2000.

[18] Mayo Clinic. LexBIG: vocabulary services for caBIG. `http://informatics.mayo.edu/LexGrid/`, 2006.

[19] D. Moher, K. F. Schulz, and D. Altman. The CONSORT statement: Revised recommendations for improving the quality of reports of parallel-group randomised trials. *The Lancet*, 357, April 2001.

[20] National Audit Office. The Cancellation of the Benefits Payment Card project. UK Stationery Office, 2000. `http://www.nao.gov.uk/publications/nao_reports/9900857.pdf`.

[21] National Cancer Institute. Cancer Data Standards Repository (caDSR). `http://ncicb.nci.nih.gov/`, 2003.

[22] Oxford University Press. Oxford English Dictionary (on-line). `http://www.oed.com/`.

[23] K. Paul. National Information Exchange Model (NIEM): Overview and Status. United States Department of Justice, January 2007.

[24] U. B. Pavanaja. Language impediments to e-governance: Problems and solutions. `http://vishvakannda.com/node/164`, 2006.

[25] A. Salminen. Building digital government. In *Proceedings of the 38th International Conference on System Sciences, Hawaii*. IEEE Proceedings, 2005.

[26] T. Stahl and M. Völter. *Model-Driven Software Development*. John Wiley and Sons Ltd, 2006.

[27] E. Tambouris. An integrated platform for realising on-line one-stop government: The eGov project. In *DEXA International workshop on eGOV*. IEEE Computer Society Press, 2001.

[28] D. Tidwell. *XSLT*. O'Reilly, 2001.

[29] UK e-Government Unit. e-Government Interoperability Framework. UK Cabinet Office, 2005. `http://www.govtalk.gov.uk/`.

[30] United States Department of Justice. Global Justice XML Data Model. `http://www.it.ojp.gov/jxdm/`, 2007.

[31] J. Vlissides. *Pattern Hatching: Design Patterns Applied*. Addison-Wesley, 1998.

[32] World-wide Web Consortium. SKOS: Simple Knowledge Organisation System. `http://www.w3.org/2004/02/skos/`, 2004.

## APPENDIX

The following figures show how metadata elements are represented in the CancerGrid metadata registry, how a metadata element and its value domain can be incorporated in a spreadsheet for data collection, and how metadata elements can be accessed during the modelling process.

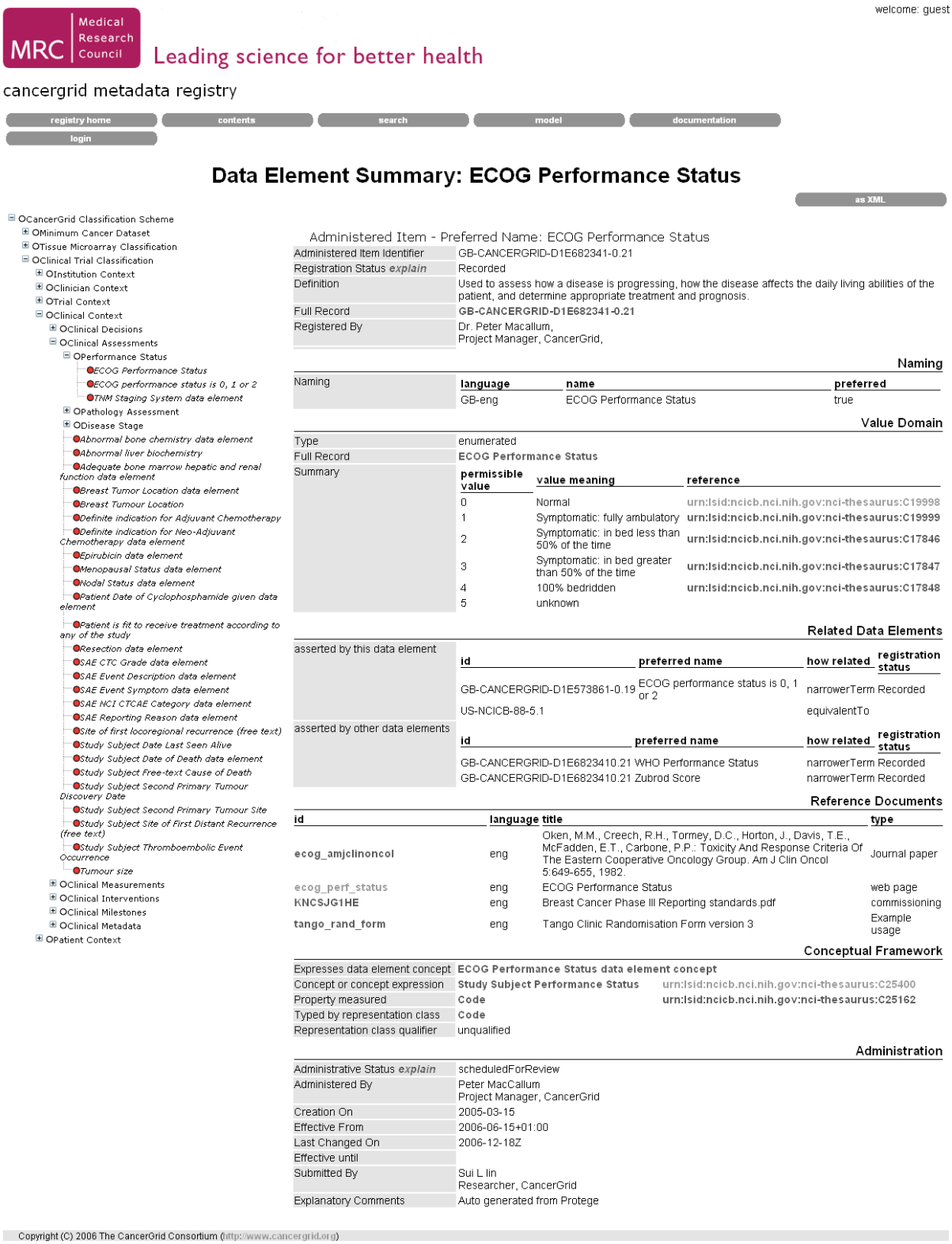


**Figure 3: A Metadata Element in the CancerGrid Registry**

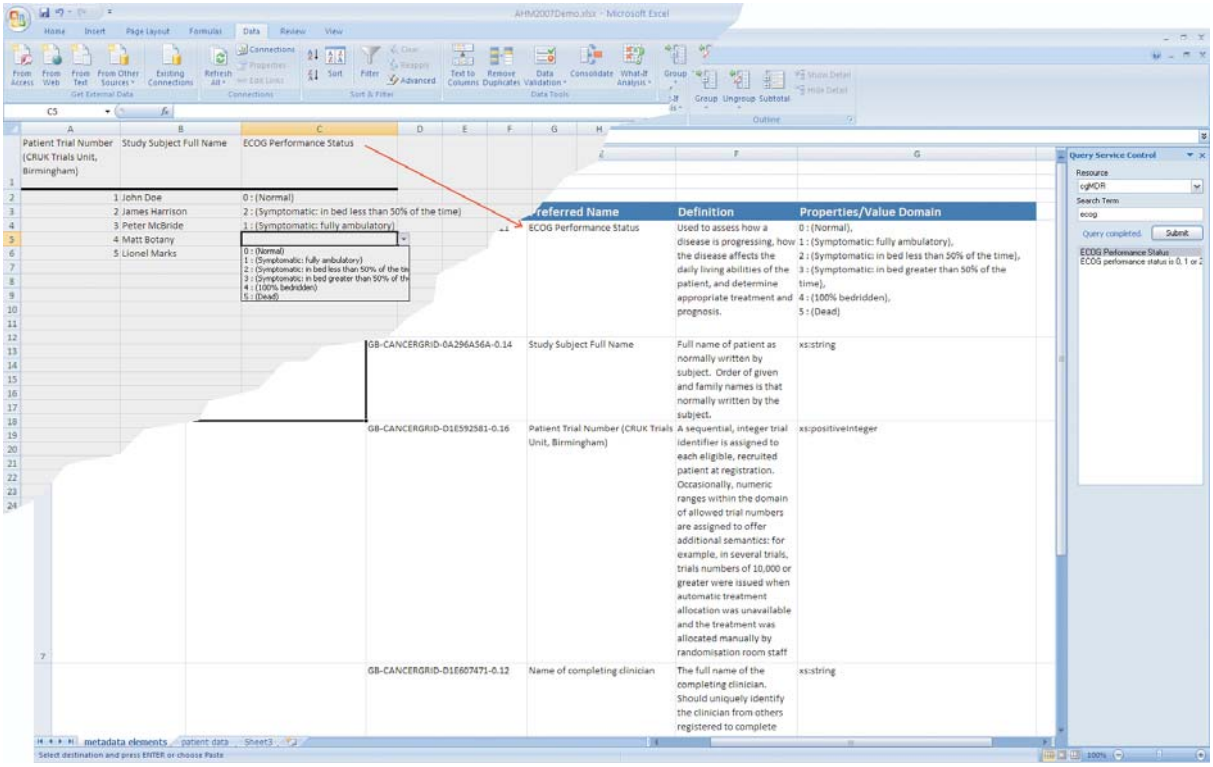


**Figure 4: Using metadata elements in Excel**

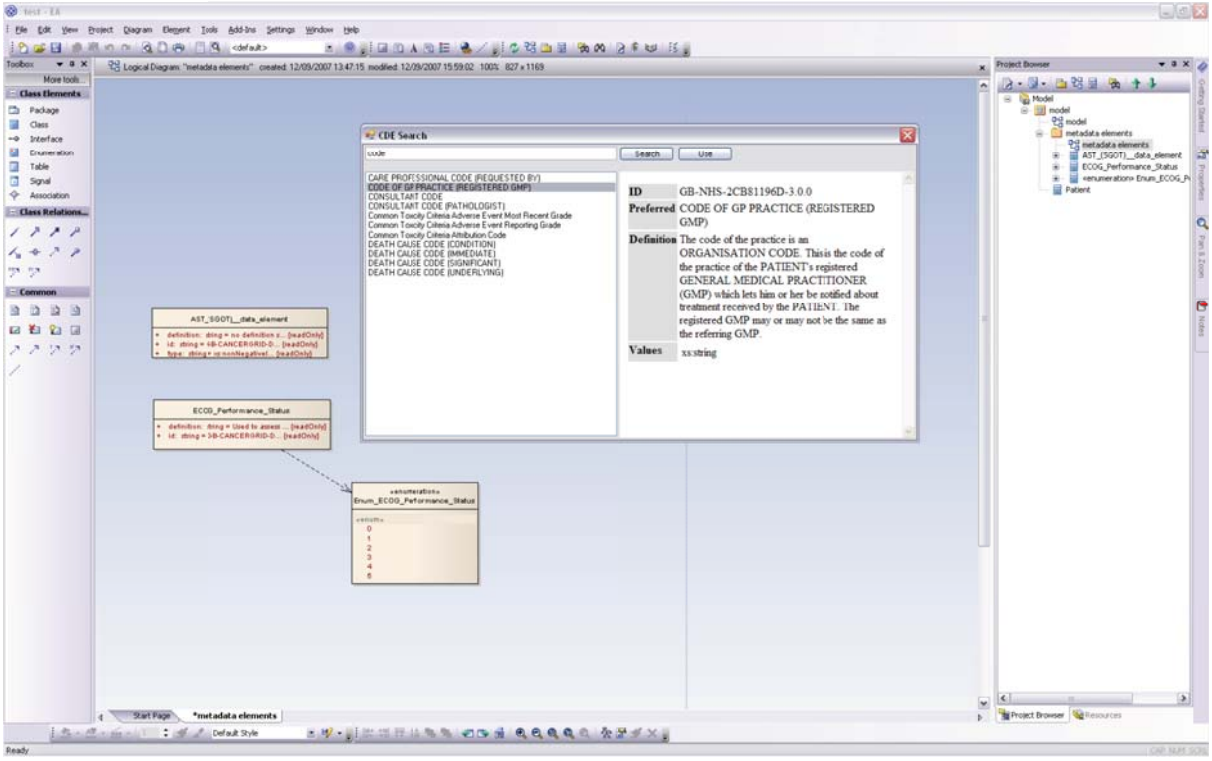


**Figure 5: Using metadata elements in Enterprise Architect**