
# A Standards-Based Registry/Repository Using UK MOD Requirements as a Basis

Version 0.3 (draft)

Paul Spencer and others

# CONTENTS


1 Introduction ..... 3

1.1 Some Terminology ..... 3

2 Current Situation (Paul) ..... 4

2.1 The Data Dictionary ..... 4

2.1.1 The Dictionary Workflow Process ..... 5

2.2 XML ..... 5

2.3 XML Management ..... 6

2.3.1 Defining XML Data Types and Elements ..... 6

2.3.2 Version Management of XML Data Types and Elements ..... 6

2.3.3 Assembling Data Types into Message Schemas ..... 7

3 Business Need (Paul) ..... 8

4 Technical requirement (Paul) ..... 9

5 Wider UK Government Requirement (Paul) ..... 10

5.1 Wider UK Requirements ..... 10

5.2 Registry/Repository Interactions ..... 10

6 (Other related requirements) (group) ..... 12

7 Proposed Solution Architecture (group) ..... 13

# 1 Introduction

This paper is the result of a joint study by Paul Spencer and three OASIS Technical Committees (TCs). The ebXML Registry TC (regrep), the Content Assembly Mechanism (CAM) TC and the e-Government TC.

The intention is to consider whether and how the standards being developed by the regrep and CAM TCs can help meet the needs of the MOD, and wider Government needs, for managing schema components. This management involves:

- registering proposed schema components as drafts;
- reviewing proposed schema components;
- registering approved schema components;
- assembling complete schemas from components; and
- managing the lifecycle of the components and schemas.

The situation at the MOD has been chosen for three reasons:

- it is relatively simple compared to the more general schema management problem;
- the requirements are well understood and there is an interim solution in place; and
- there is a need to replace the interim solution in the medium term as the number of schema components grows.

This study relates only to those data with a security classification of "Unclassified".

## 1.1 Some Terminology

A registry stores information about objects (i.e. metadata).

A repository stores the objects themselves.

The Defence Data Repository (DDR) is an existing data dictionary held in a database. Using the definitions above, this is a registry.

Accord is the system that holds the data dictionary and provides workflow management for reviewing and approval.

The Government Data Standards Catalogue is a data dictionary operating across UK Government and holding information about data items widely used across Government. See http://www.govtalk.gov.uk/gdsc/html/.

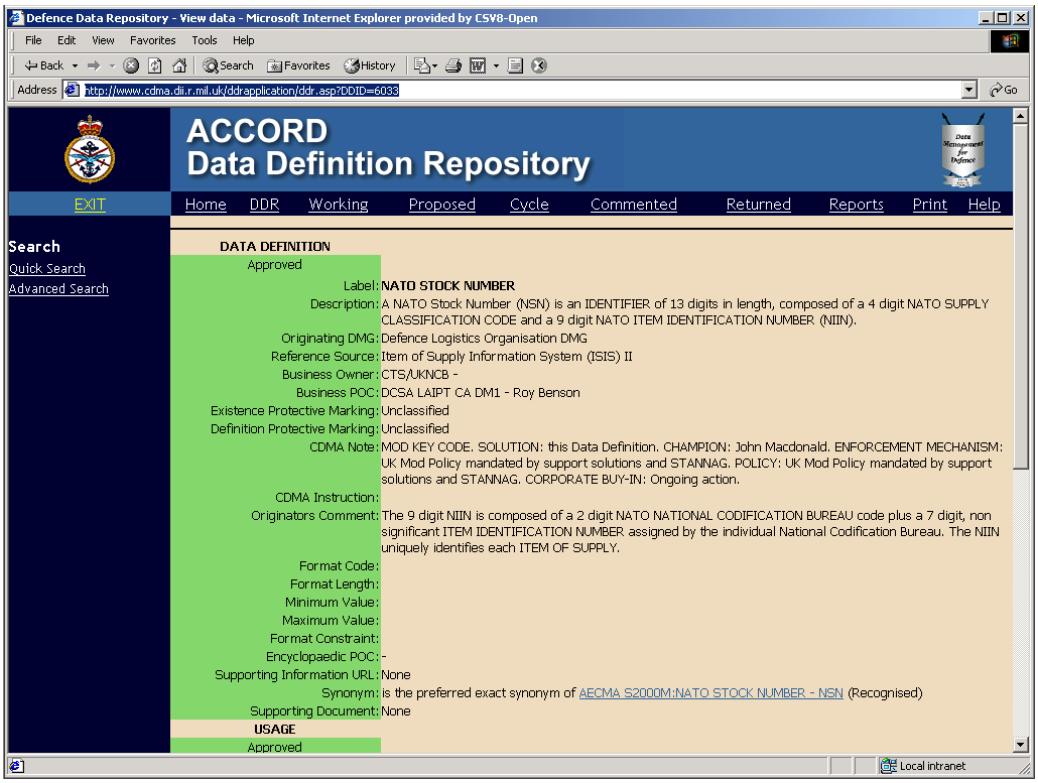
# 2 Current Situation (Paul)

## 2.1 The Data Dictionary

The MOD has a sophisticated data dictionary developed in-house using a SQL Server database. The dictionary is referred to as the DDR, and the System on which it runs is called Accord. This dictionary currently has over 4000 entries, with an expectation that it will grow substantially, possibly to around 100,000 entries.

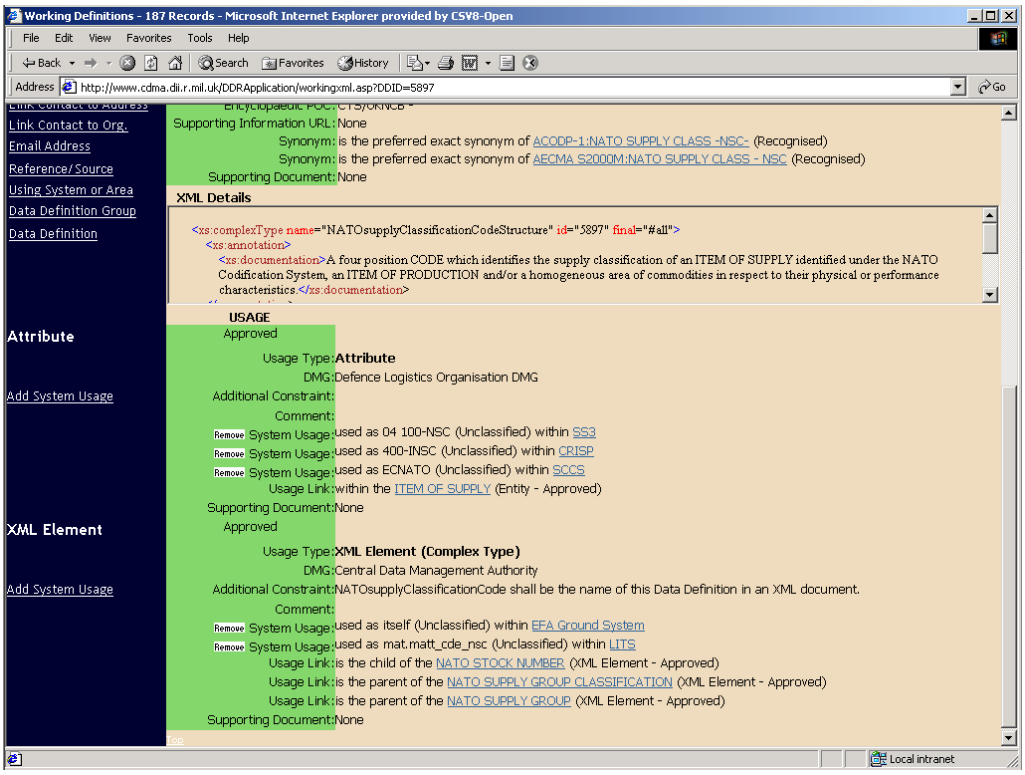
Accord includes support for proposing, reviewing and signing off dictionary entries. The following screen shots give a feel for the system.

This screen shows the top part of the definition of a NATO Stock Number. (Note that the heading is incorrect - it should read Defence Data Repository rather than Data Definition Repository.)




The next screen shot shows the remainder of that definition. This includes the XML Schema definition of the dictionary entry as a complex data type.




### 2.1.1 The Dictionary Workflow Process

Any logged-in user can create a data definition. This is given the status of "Working". The creator of the data definition, and his or her nominated group, can then amend it as they choose. When they are satisfied, they change the status to "Proposed". Control of the data definition is then passed to a central authority for review, which may include minor amendment. If major amendments are required the data definition is returned to the originating group.

Once the central authority is content, it changes the status to "Candidate". Candidate data definitions are batched together and passed out to all users for comment in a two-week long cycle. Data definitions in an open cycle cannot be amended.

When comments are complete, normally after two weeks, control is returned to the central authority which will action any agreed amendments and attempt to resolve any outstanding issues. If issues cannot be resolved, the data definition is returned to the originating group.

Where agreement is reached, the central authority changes the status to "Approved", or more rarely "Rejected". Only approved data definitions are mandated for used in the MOD.

Accord supports this workflow, providing mechanisms for managing and tracking the process.

## 2.2 XML

The MOD is starting to use XML widely, and has adopted a set of policies in use throughout the organization. It is some of these policies that simplify the management of XML artifacts within the MOD, allowing the present management

system and making this a useful case study for use of OASIS regrep and CAM standards.

Those policies most relevant to this study are:

- XML will be the standard for exchanging information within the MOD and between the MOD and external partners.
- The XML will be defined by schemas defined using the W3C XML Schema recommendation.
- Messages will comprise a standard MOD header and a document payload.
- These schemas will be based on definitions held in the DDR.
- All objects must be defined as data types. Anonymous data types must not be used and elements are only declared globally where they can be the root element of a document payload.
- Namespaces will be defined only for different security classifications of information.
- Where messages must be exchanged with parties not using these standards, a translation will occur at the boundary of the MOD domain so that only XML conformant to the policies exists within the MOD domain.

A result of these policies is that all objects within the MOD are declared as XML data types. In practice, these data type definitions are currently held within a single text file, which acts as a repository. Each message type is defined in a separate schema document. This document will just include the standard MOD header with a single global element declaration.

Since this study is only dealing with unclassified data, it is dealing with XML artifacts in a single namespace.

## 2.3 XML Management

There are three main issues of XML management in the MOD:

- proposing and approving XML data types and elements;
- version management of XML data types; and
- assembling data types into schemas for message types.

### 2.3.1 Defining XML Data Types and Elements

Once a data definition is approved, the associated XML data type is defined centrally. There is currently no formal approval mechanism for this.

### 2.3.2 Version Management of XML Data Types and Elements

The version management of data types is managed simply through the use of a `Version` attribute in data types and elements that reference the data type. Since this attribute is not defined in XML Schema, it is created in an MOD namespace. The example below shows the declaration of version 1.0 of the (global) complex data type `NATOstockNumberStructure`. This type comprises a sequence of two elements, `NATOsupplyClassificationCode` and `NATOitemIdentificationNumber`. Since these are declared locally, the relevant data types must be referenced. In each case, we are using version 1.0 of the data type.

```
<xs:complexType name="NATOstockNumberStructure"
```

```
                id="5897"
                mod:Version="1.0">
  <xs:sequence>
    <xs:element name="NATOsupplyClassificationCode"
                type="NATOsupplyClassificationCodeStructure"
                mod:Version="1.0"/>
    <xs:element name="NATOitemIdentificationNumber"
                type="NATOitemIdentificationNumberStructure"
                mod:Version="1.0"/>
  </xs:sequence>
</xs:complexType>
```

In this way, later versions of any of these data types can be defined, but the definition of the `NATOstockNumberStructure` will only be changed if this is done explicitly by changing the version numbers in the element declarations.

Global element declarations are handled in exactly the same way, effectively giving versioning of message types.

### 2.3.3 Assembling Data Types into Message Schemas

Assembling the elements and data types into a schema specific for a message type is a simple iterative process. Starting at the root element of the message payload, each required data type can be placed into a schema document. In each case, the correct version of the data type must be selected.

Currently, this is done as a two stage process using XSLT. The first stage collects the correct version of each data type from the repository document each time it is referenced. The second stage removes the duplicates. It is an error if two versions of the same data type are required.

# 3 Business Need (Paul)

Currently, Accord acts as a registry for schema components, and the text file acts as a repository. Although this works well at present, there are some obvious limitations:

- The solution is not scalable - the text file will grow both with new data types and new versions of existing data types.
- When a new version of a data type is created, it is a manual process to update all related types.
- There is no means of integrating the registry and repository with others in the UK Government domain, the international military domain or other domains of interest.

The main business need is therefore to preserve the functionality of the existing system while addressing these three issues.

Additional requirements are as follows:

- In future, people are likely to propose XML data types with their data definition, so a mechanism is required to approve this as is done with the current definitions. This approval could be simultaneous with that for the data definition, or could follow it.
- It must be possible to generate metadata based on the UK Government metadata standard (http://www.govtalk.gov.uk/schemasstandards/metadata_document.asp?docnum=832).
- The resulting system must be scalable and robust. Eventually, there could be around 100,000 type definitions stored.
- When a new version of a global type declaration is produced, the current situation whereby there are no accidental knock-on effects on other declarations must remain. However, it should be as easy as possible to incorporate the updates.
- It must be possible to identify those declarations that use old versions of the definition, all the way up the tree to a payload root element declaration. Possibly, all the intermediate declarations could be created so that only the root element declaration is changed manually.
- It must be possible to perform a "what-if" analysis, whereby the impact of a planned change or deletion can be assessed.
- It must also be possible to identify unused definitions so that they can be purged.
- It must be possible to integrate the registry and repository to others as required.

# 4 Technical requirement (Paul)

- The system must run on existing MOD infrastructure. This is Based around Microsoft Windows and SQL Server.
- The system must maintain integration with the DDR. This can be either by integrating with Accord (which can be modified to suit) or replacing it.

# 5 Wider UK Government Requirement (Paul)

These requirements are common across much of UK Government (and, indeed, governments and industry in general). There is therefore scope for compatible systems to be used elsewhere, potentially allowing benefits to be derived through the use of federated systems.

UK Central Government already holds a data dictionary called the Government Data Standards Catalogue (GDSC). This is small compared to dictionaries in individual departments, but has the scope to grow bigger. It is currently held as an XML document with tools to render it as a PDF document or a set of HTML pages.

Each definition in the catalogue has a representation using W3C XML Schema, related representations being grouped into one of a set of schema documents. Note that the MOD policies do not apply to these documents. There is a desire to move this into a regrep environment.

## 5.1 Wider UK Requirements

- The registry would need to be compatible with the UK Government Change Control Procedures (http://www.govtalk.gov.uk/documents/Change-control-v1_0-2002-09-24.pdf).
- The registry and repository would have to be accessible via the UK GovTalk web site (http://www.govtalk.gov.uk/).
- Whilst the MOD requirement is just to store XML Schema artifacts, the wider need will include storing Schematron and possibly other schema languages.
- Whilst the MOD requirement is for simple namespaces, the wider requirement will require each component to have its own target namespace (or no target namespace in some cases).
- It should interface with standard tools, such as XML Spy, to allow easy access to the repository.

## 5.2 Registry/Repository Interactions

There are several ways in which the MOD could interact with such a repository.

In terms of access to remote schemas, it could:

- duplicate central repository items in its own repository (perhaps modifying them to fit into the MOD policies); or
- access the remote repository items to create its own message schemas; or
- access the repository "live" as required.

The last of these options is not a good idea as it moves control of change and service performance away from the MOD. In practice, the MOD would choose the first option, although other organizations might prefer the second. In either case, a means is required to keep information coordinated.

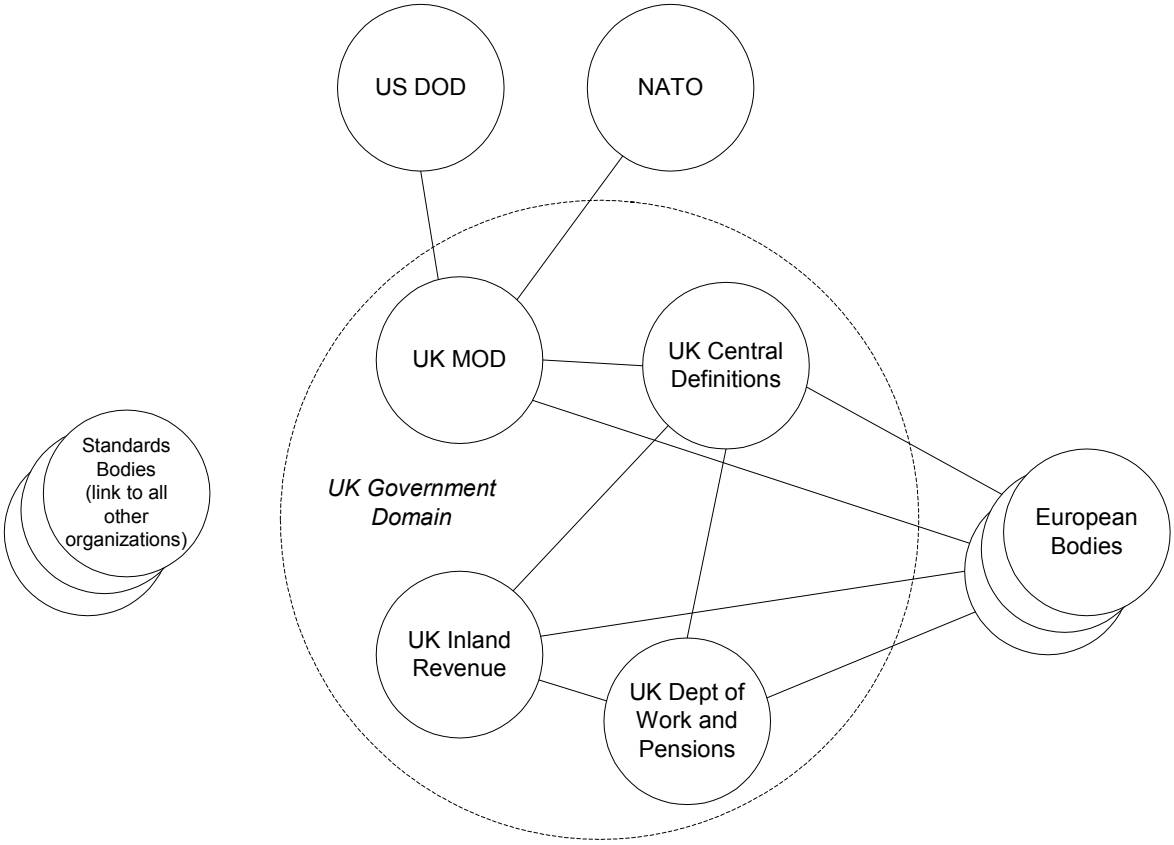
In terms of publishing its own definitions, the MOD could:

- publish to the central repository; or
- make a part of its own repository available to others.

There are therefore two basic models of distributed information - a central repository of shared items, with individual public sector organizations uploading and downloading as required or a fully distributed model with the repository distributed over multiple systems.

The discussion above related the MOD registry/repository to Central Government. This model could clearly be used across a closed government environment. However, the public sector in general requires access to other XML definitions, such as those used in UBL. This is the main reason for using open standards and working towards a common architecture. The diagram below shows some of the interactions that could be aided by a common, standards-based, architecture.



# 6 (Other related requirements) (group)

# 7 Proposed Solution Architecture (group)