Cabinet Office
Office of the e-Envoy

Towards e-government

UK Online – Information Architecture – Overview

Document Type: Overview / Guidance

Version: 1.0
Status: Agreed

8 March 2001

Document Printed 12/03/02

Document Control

Contributors		
Name/Position	**Organisation**	**Contact Details**
Paul Spencer	Boynings/IR	paul.spencer@boynings.co.uk
Chris Gahan	BT	Chris.gahan@bt.com
Ian Jones	BT BS	ian.c.jones@bt.com
Phil Bull	BT BS	phil.bull@bt.com
Steve Towndrow	BT ExaCT	steve.towndrow@bt.com
Tony Fletcher	BT ExaCT	tony.am.fletcher@bt.com
Paul Muschamp	BT Group Technology	paul.muschamp@bt.com
David Hockley	BT Syntegra	dave.hockley@syntegra.bt.co.uk
Kulbir Kaur	BT Syntegra	kulbir.kaur@syntegra.bt.co.uk
Neil Glover	CCTA	neil.glover@ccta.gov.uk
John Ross	CCTA/SECSTAN	ross@secstan.com
Anwar Choudhury	Office of the e-Envoy	achoudhury@citu.gsi.gov.uk
Mike Hudson	Office of the e-Envoy	MHudson@citu.gsi.gov.uk
Paul Ukpai	Office of the e-Envoy	pukpai@citu.gsi.gov.uk
Sebastian Madden	Office of the e-Envoy	smadden@cabinet-office.x.gsi.gov.uk
Graham Beaver	Compaq	graham.beaver@compaq.com
Nik Webster	Compaq	nik.webster@compaq.com
Mike Burston	DSS	
Peter Desmond	DSS	p.desmond@new100.dss.gsi.gov.uk
Ken Clarke	ECSoft	Ken.Clark@ecsoft.co.uk
Richard Taylor-Carr	ECSoft	Richard.Taylor-Carr@ecsoft.co.uk
Simon Slocombe	ECSoft	simon.slocombe@ecsoft.co.uk
Rod Mountney	EDS	rod.mountney@eds.com
David Treder	environment-agency	david.treder@environment-agency.gov.uk
Andrew Coote	ESRI(UK)	acoote@esriuk.com
Clare Samways	ESRI(UK)	csamways@esriuk.com
Paul G Butler	IBM	paul_butler@uk.ibm.com
Nig Greenaway	ICL	nig.greenaway@icl.com
Rosi Somerville	IDeA	Rosi.Somerville@IDEA.gov.uk
Steven Brandwood	IDeA	steven.brandwood@idea.gov.uk
Simon Berlin	LB Lambeth	SBerlin@lambeth.gov.uk
Matt Bishop	Microsoft	matbisho@microsoft.com
Linda Williams	Nineveh Consulting	linda.williams@dial.pipex.com

Abstract

The UK Government is establishing the means for citizens and businesses to be able to transact business with the Government electronically. This will use a variety of delivery mechanisms such as the Internet, public 'kiosks', mobile equipment, and interactive television. This series of documents specifies the general information architecture for the overall system. This document provides an introduction to this information architecture. In particular it specifies its scope and the general design considerations.

Current Version

Date	Version	Status	Editor/ Author	Comment
8 March 2001	1.0	Agreed	A Fletcher, BT	Approved by Interoperability Group

Authorisation

Name	John Borras
Organisation	Office of the e-Envoy
Date	8 March 2001

Change History				
Date	**Version**	**Status**	**Editor/ Author**	**Comment**
14 December 2000	*0.2*	Draft	A Fletcher, BT	Settled on a definition of a schema in sub-section 3.1 and on mapping rules from Information Architecture to message in sub-section 4.3.

Contents

Figures

1. Scope of the UK Online Information Architecture

The scope of the UK GIF includes the interactions between:

- UK Government Department and other UK Government Departments
- UK Government and wider public sector
- UK Government and Foreign Governments (e.g. UK/EC, UK/US etc)
- UK Government and Businesses (world-wide)
- UK Government and UK Citizen

UK Government includes Central Government Departments and their Agencies. The wider public sector includes Local Government, Devolved Administrations, Non Departmental Public Bodies and the National Health Service.

Figure 1 illustrates the overall system design. Citizens, and other 'users', interact with and receive services from a Web portal. Initially this will be the UK Online portal, but competitive portals may follow on later and there are also plans for other types of portal such as ones serving small business. Delivery is not limited to the Web, but will be extended to other channels, such as public kiosks (which may actually be just web browsers), mobile phones and interactive television. These various 'portals' will serve content directly (or indirectly via 'outer' local, regional, commercial or other portals), and will communicate with local and central government systems via a gateway system to perform transactions (such as notifying a change of address). The overall system is not limited to citizen to department communication. It can also be used by applications, for department to application (citizen or business), and for department to department communication. Authentication will be required before transactions are undertaken.

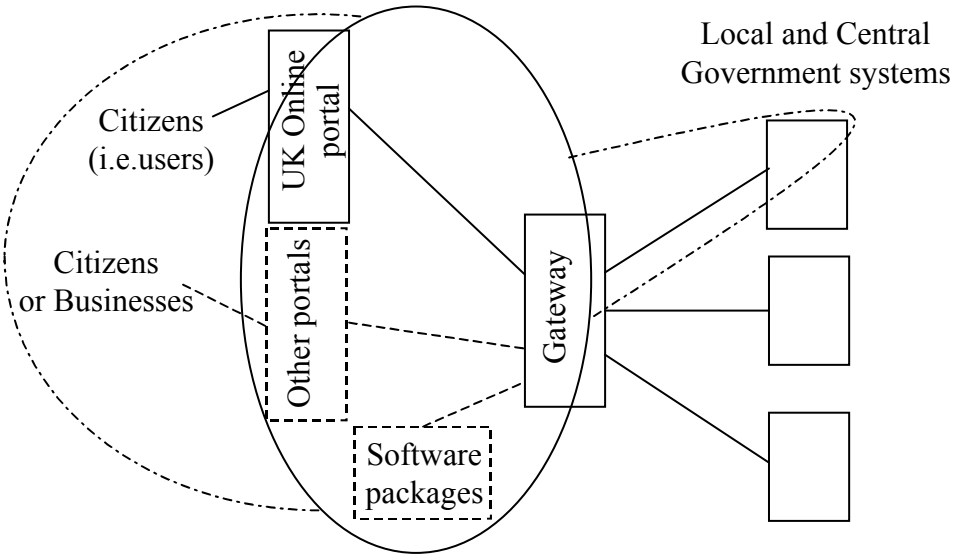

Figure 1 Overall system design for UK Online

The other documents in the UK Online – Information Architecture series provide data schema in UML and XML form for various 'services' and composite pieces of information of interest. These are built from the Government Data Standards (GDSs) for fundamental data items that are specified, in detail, in the eGovernment Interoperability Working Group (eGIWG) Data Standards Catalogue [6]. Taken together these schema build into larger schema and hence effectively into a complete information architecture. The solid circle illustrates the intended scope of this information architecture. It applies particularly to the portals and the gateway, as these are new systems that should be built specifically to comply with this information architecture. The communication between the portal(s) and the gateway shall use XML structured messages that have their elements drawn from this information architecture. As also shown in figure 1, the designers of software packages (called

'applications' above), can opt to provide an interface to the gateway that is compliant with the Information Architecture and conformant to specified UK Online message specifications.

The Information Architecture specified in this series of documents is applicable to the portal to user interface, in addition to any portal to portal interfaces and the portal to gateway interface. Where the portal determines the interface it is recommended that it comply with this Information Architecture. Where the user end determines the interface, as would be the case if the user were using an application package, it is recommended that this Information Architecture be adhered to, so that confusion over data semantics is avoided. However, in this later case the user – portal interface may depart from compliance with this Information Architecture by agreement with that portal provider.

> *Note:* The Information Architecture for the interfaces that are internal to the overall system, such as those between a portal and the gateway and between the gateway and a departmental system, strictly only need to convey the semantics for each value of a data item. Thus 'Y' and 'N' would do as distinguished values that convey the semantics of 'yes' or 'no' respectively. For the system to outside world interfaces, such as that from a portal to a user, it may be necessary to extend the Information Architecture in various ways including to allow multiple values with the same meaning (e.g. 'Y', 'Yes', 'YES' and language variants to convey the meaning 'yes'). Although 'radio buttons' and 'drop down lists could be used to present options to a user in appropriate language and yet have a single value per semantic on the interface, use of this approach alone could over constrain the user interface design.

At the other end of the system lie the various central and local government systems. These exist and will conform to their own pre-existing data conventions. It is unrealistic to expect all these to change. The information architecture specified in these documents shall, however, be regarded as the target information architecture that these systems should be migrated to over time (i.e. when they are replaced by a new system), and should be regarded as the common 'language' of the overall system. This is illustrated in figure 1 by the dotted arc that extends the circled scope of application. The practical implication is that, where the data definitions used in the local and central government systems do vary from that in this information architecture it will be necessary for that system, or the gateway, to perform a translation function.

2. Notational Basis of the Schemas

The UK Government UK Online information architecture and message specifications are being developed under the eGovernment Interoperability Framework (eGIF [1]) as a set of UML [2] models and corresponding XML Schemas. The XML Schemas adhere (currently) to the April 2000 draft of the XML Schema Definition Language (XSDL) specification of the World Wide Web Consortium (W3C). This specification is in three parts [3, 4 & 5].

Later, the UK Online specifications will be updated to comply with the full release version of the W3C XML Schema Definition Language (XSDL) specification.

This specification allows many options for how schemas can be defined. A companion Guidelines document [7] defines the best practice to be used within UK Online.

Reference is made to the Data Standards Catalogue (DSC) [6] developed by the Data Standards Working Group (which represents several Government Departments). This group has been brought under the eGIF umbrella. The DSC might therefore change its name and/or format.

3. Schemas

3.1. What is a Schema?

A schema is the definition of allowable content in a given context to enable both sender and recipient to share a common understanding of the meaning of a message. They are intended to document their own meaning, usage,

and function through a common vocabulary. Thus, XML Schema Structures can be used to define, describe, and catalogue XML vocabularies for classes of XML documents by using mark-up constructs to constrain and document the meaning, usage and relationships of their constituent parts. Schema constructs also provide for the specification of additional information such as default values.

3.2. Schema Types

There are likely to be several distinguishable types of schema for different purposes and generated by different means developed for UK Online. The most common will be:

UK Online definitions of common data types and elements for re-use elsewhere (for example, the definition of an `Address` data type or a `NationalInsuranceNumber` element). Individual basic data items are specified in the Data Standards Catalogue (DSC) [6]. These are composed into useful structures driven by requirements for specific 'services', such as notification of a change of address. These individual mini schemas build into the overall (static) Information Architecture. The UML representation in these schema documents is in the form of Class diagrams.

UK Online message schemas created manually based on common elements without the use of an external repository.

UK Online message schemas automatically generated from a metadata repository.

The UML representation in message schema documents is in the form of restricted class diagrams, where each complies with a Class from the static data (Information Architecture) Class diagrams or a restriction or specialisation (i.e. sub-class) of such a class. Message schema may restrict the multiplicty and class attribute values relative to the Information Architecture Class diagram (refer to sub-section 4.3).

The term *schema* on its own (or data schema) refers to a view of data and the way that each 'piece' of data relates to other 'pieces'. Taken together these schema aggregate into an overall 'Information Architecture. The term *message schema* refers to a schema that defines a single message. For example, the Inland Revenue might define a (data) schema called SA100 for a self-assessment tax return, while the UK Online Gateway team might define schemas to send self-assessment tax return and respond to an attempt to file a self-assessment tax return. Each of these would be a message schema.

4. Unified Modelling Language

4.1. Introduction

The Unified Modelling Language (UML [2]) is an object oriented modelling language that has been standardised by the Object Management Group (OMG). It currently consists of nine different graphical modelling techniques that have been related together, hence the 'unified' part of the name. Each technique has its own uses. Currently it is envisaged that the following subset of these techniques will find use in the UK Online documentation.

Use Cases are applicable to the high level specification of services.

Class diagrams are used to specify the information architecture schemas contained in this series of documents. Message schemas also use class diagrams. These will copy or specialise the classes from the the information architecture schemas. They may also restrict the multiplicity and class attribute values over the equivalent classes in the information architecture (but they may not be less restrictive).

State (transition) diagrams and message sequence charts (of which there are three types: collaboration, sequence and activity) are used to specify behaviour, such as the sequencing of messages. State (transition) diagrams are used to specify the complete permitted sequencing of messages, while message sequence charts are used to illustrate the typical sequencing of messages.

UML has the advantage of being in the nature of a formal, well-specified language (though it is not properly formal in the mathematical sense). However, having graphical notation, it is relatively easy to understand. This is one of the main reasons for using it in these specifications. The intent is that readers of these series of

specifications will easily be able to understand the basic intent, even if they will need to know UML in detail to understand all the nuances.

UML is a modelling technique that is useful for the sharing of ideas. It does not include a syntax that can be used directly for the storage and interchange of data items. This should be regarded as a strength, in that it is possible to map UML class and object diagrams to various transfer syntaxes. For UK Online we have chosen to map the UML to XML schema language [3, 4 & 5], which in turn specifies the detailed use of XML for the storage and interchange of data.

4.2. UML notation

The notations used for each type of UML diagram differ in detail, although there is a strong family resemblance between them all. There are several books on the market, which describe the notation and show how it can be used for modelling various sorts of things. This series of documents on data schemas just use Class diagrams. The closely related series, which specifies the various messages used within UK Online also use class diagrams. Figure 2 is a contrived example of a simple class diagram of the sort used to specify the data schemas.

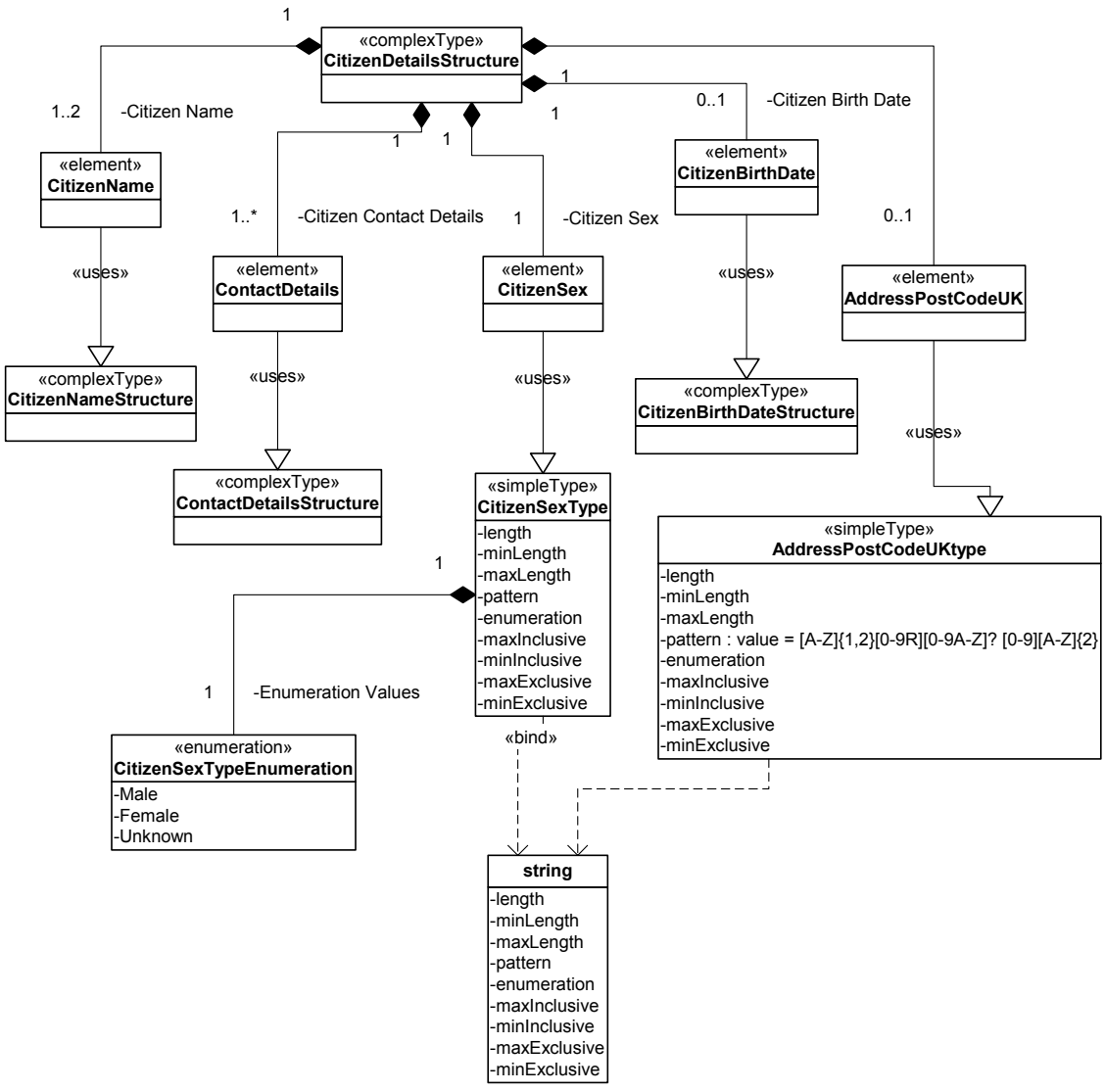

Figure 2 An example UML Class diagram

Each diagram has a single class at its top most level. In this case it is the class named *CitizenDetailsStructure.* A rectangle, which can have two, or more, divisions, denotes a class. The top part contains the name of the class (plus sometimes some additional information such as *standard = ISO 639* where standard is the type of the extra

information and ISO 639 its value - these are known as tagged values). The second division contains a list of the class attributes. The third division is not currently used in the schema diagrams, but where it is shown it contains the operations that can be performed on that class (such as read, delete, etc.).

The filled diamond shape leading from the class means that the class next to it, *CitizenDetailsStructure* this case, is composed of all the classes that have such associations leading from this class. So this particular class is composed of four other classes. If an open diamond is used then it means that the objects of the classes can exist on their own, whereas the solid diamond means that if an object of the top class ceases to exist then so do all the objects that compose it. Taking the composition relationship from *CitizenDetailsStructure* to *CitizenName* the one means that *CitizenName* is associated with exactly one *CitizenDetailsStructure*. The 1..2 at the other end means that there can be one or two *CitizenName* components to a *CitizenDetailsStructure.* Similarly for the other composition relationships - 1..* means that there must be at least one, but can be as many more as you like, 0..1 means that an component is optional it may be omitted altogether or may be present just once, and similarly for other possibilities. The triangle an a relationship line means that the class at the is a sub-class, or a specialisation of the class at the triangle end, so for instance a 'UK Address' may be a specialisation of a more general 'Address' class.

The part of the schema down to the classes designated as <<element>> is the main part of the Information Architecture part of the schema. Below that the schemas are more specific to the XML realisation. So the class *CitizenName* is represented in the XML realisation as an XML element. The more generic class *CitizenNameStructure* gives the type of the values of this element. These can have a complex structure themselves, as in this case, or be a simple datatype as in the case of *CitizenSexType*. latter simple datatype is itself a special use of the more general string type and also takes a fixed set of values, known as an enumeration. In this case the values are 'Male, Female' and 'Unknown'. Every branch of the diagram ends in a simple datatype where this is not the case it means that the complex type is expanded until it ends in simple types on another page of the schema.

The *AddressPostCodeUKType* contains what is known as a pattern. The notation and conventions for these are found in the draft W3C Recommendations on XML Schemas [3, 4 & 5]. Patterns enable the XML part of the system to apply a graeter level of validation than just checking the length of an element value (for instance). However it is not practicable to use patterns for very detailed checking or for meaning in context – the applications making use of the data values will still need to apply these business rules. For instance in this case it is not really practicable to express in the pattern rules of the nature if this letter appears here then that letter can not appear there. Also the XML ‘layer’ can only check that the value corresponds to the specified format, the application has to check that the postcode actually exists and is the relevant one in the context. The pattern in this example is:

[A-Z]{1,2}[0-9R][0-9A-Z]? [0-9][A-Z]{2}

This particular pattern means that the first 1 or 2 positions ({1,2}) must be the capital letters A to Z ([A-Z]), the next position must be a single digit or the letter R ([0-9R]) then the next position may be omitted, signified by the ?, or may be a number or a capital letter ([0-9A-Z]?), the next position is a space a single digit and this is followed by exactly 2 capital letters ([0-9][A-Z]{2}). Thus the string IP5 3RE will be passed by this pattern check, but the string 5IP 3RE would fail.

4.3. Message Schema Class restriction rules

Two types of schema are supported by GovTalk - entity and message schemas. Messages are composed of a set of entities. Entities are defined within (and collectively define) the information architecture. Entities will usually participate in a hierarchy e.g. a Postcode is part of (contained in) a postal address which is, in turn, part of a comprehensive address (that may also contain a geographic address).

In the above example, a message schema may be generated that uses comprehensive address, postal address, geographic address or postcode classes.

All classes contained within the used class are also taken into the message schema - e.g. if Postal Address is used then the (encapsulated) Postcode will also be picked up.

The message schema may specialise (override/restrict) the elements it contains BUT ONLY WITHIN the constraints set by the entity schemas. E.g., if postal address describes sub-entity (such as mailsort code) as

optional (occurrence 0..1), the message may require that mailsort code is optional (Occurrence 0..1), never present (Occurrence 0), or mandatory (Occurrence 1). This does not invalidate the information architecture, but reflects the needs of this message.

In general the message lower bound must be equal to or higher than the information architecture lower bound and the message upper bound must be equal to or lower than the information architecture lower bound.

Other specialisations may also be valid (e.g. restricting a pattern to some subset of that specified for the entity in the information architecture). For example, if putting together a schema for a purely local service in Reading, it may be required to ensure that, whilst maintaining the standard format, all postcodes should begin with 'RG'.

Message class diagrams shall take classes from an Information Architecture Schema. All classes 'lower' than that class shall also be taken into the message schema (that is all classes that are 'children of the class taken). However, the message schema may further specialise a class. In this case a note should be added to indicete the original class that has been specialised (or use tagged values {parent_class = 'xxxxx'} ?). Also the message schema may restrict the values of class attribute values and multiplicity within the range allowed by the information architecture class diagram. For instance, if the Information Architecture class diagram permits a multiplicity of 0..1 then in the message that class may be absent altogether, mandatory (always present), or optionally present.

5. Compliance

The Information Architecture specified in this series of documents is specified in UML [2] and XML [[3, 4 & 5]] syntaxes. These two representations are intended to be semantically equivalent. Any perceived difference in the UML and XML representations is a potential specification error and should be notified to the maintenance authority for that document. The use of these two representations does not preclude the use of other representations in future versions of this series of documents, nor in protocol specifications.

Other specifications that form part of the UK Online document set shall comply with this specification. This specification provides guidance to authors of UK Online specifications and implementers of systems, and places requirements on other documentation, particularly those that specify schemas. This specification does not contain any provisions that implementations are required to conform to.

Note: Other UK Online specifications may contain provisions that implementations are required to conform to.

6. Abbreviations

Abb	Definition
CITU	Cabinet Office Central Information Technology Unit
DSC	Data Standards Catalogue
eGIF	Electronic Government Interoperability Framework
FTP	File Transfer Protocol (an Internet protocol for managing and transferring files)
HTTP	Hypertext Transfer Protocol
OMG	Object management Group
UML	Unified Modelling Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSDL	XML Schema Definition Language

7. References

	Document References
[1]	The eGovernment Interoperability Framework http://www.govtalk.gov.uk/egif/home.html
[2]	OMG Unified Modelling Language Specification, Version 1.3, June 1999 ftp://ftp.omg.org/pub/docs/ad/99-06-08.pdf
[3]	XML Schema Part 0: Primer, W3C Working Draft, 7 April 2000 http://www.w3.org/TR/xmlschema-0/
[4]	XML Schema Part 1: Structures, W3C Working Draft, 7 April 2000 http://www.w3.org/TR/xmlschema-1/
[5]	XML Schema Part 2: Datatypes, W3C Working Draft, 7 April 2000 http://www.w3.org/TR/xmlschema-2/
[6]	Data Standards Catalogue, Version 02, 29 February 2000
[7]	UK Online – Information Architecture - Guidelines
[8]	

END OF DOCUMENT