

# XML and Objects - the future for e-Forms on the Web

Anders Tornqvist  
Chris Nelson  
Mats Johnsson  
Dimension EDI Ltd.

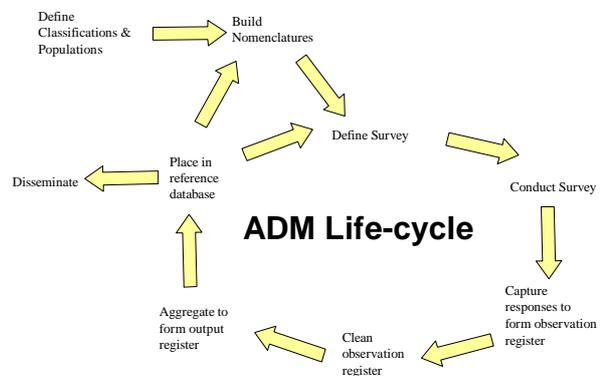
## Abstract

Many organisations are using the Web for electronic commerce or purely for the collection and dissemination of information based on forms and tables. For many of these organisations a bespoke Web design with specialised Java Script is what they want. However, for organisations not wishing to make this investment, a flexible system based on a standard XML DTD could fit the bill. Fully functional E-forms can be prepared and maintained for a fraction of the price of a bespoke Web design. Based on a combination of the knowledge of the real meaning of data and metadata, XML and the Web, and object oriented technology, there is a new way to develop e-forms and to integrate them with applications.

## The Problem Domain

Most organisations collect data, check it, build retrieval mechanisms for it, and sometimes they publish it. This cycle comprises an often necessary set of steps that enable the organisation to manage its primary business, be it retailing, manufacturing, banking, insurance etc. For statistical organisations or market research companies, where collection, aggregation and dissemination of data is their business, this cycle is of primary importance and such organisations spend a lot of time making sure they do this effectively and efficiently.

In the Object Management Group (OMG), this cycle is known as the Analytical Data Management (ADM) life cycle.



## The ADM Life Cycle

The specification of object standards that support this life cycle is being actively pursued by the Analytical Data Management Special Interest Group at the OMG, which is now specifying CORBA Domain Standards. Parts of this life cycle are used by most organisations.

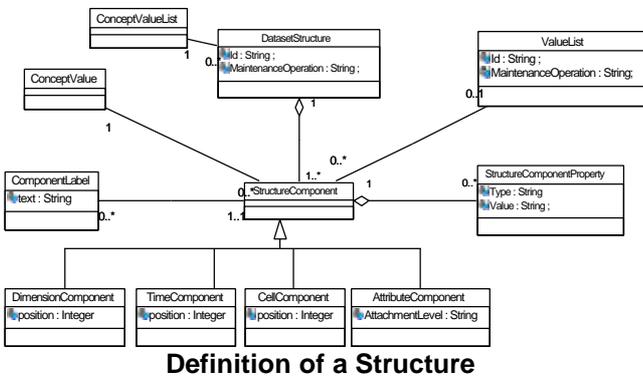
This paper concentrates on the first part of the life cycle - collecting the raw data. Raw data can be anything. Basically, if there is a form involved, even an electronic form, then it is probably raw data. A statistical questionnaire is raw data, a sales report made by a salesman is raw data, a product order form is raw data.

The model and supporting software described in this paper covers the “Define Survey”, “Conduct Survey”, and “Capture Response” parts of the life cycle. The model itself has been developed by a group of national statistical institutes and central banks in Europe, and it is being fed into the work of the ADM Sig at the OMG. The initial intention of the European group is to define an XML DTD, but in such a way that systems can migrate easily to use native object technology in the future. Questionnaires are actually forms by another name, but quite complex forms because they must take account of:

- validation of input (not just simple numerics, but “look up” tables, even remote object processing)
- navigation (answer “yes” to box 11 then “freeze” input for boxes 12-17 and place mouse pointer at box 18)
- multi-linguism (not only the language on the form but also the language of labels for codes (e.g. 1=Yes, Ya, or Oui))
- calculation

## The Object Model

A good XML DTD must be derived from an object model. This will facilitate the integration of object technology such as CORBA, and will ease the program development in object languages such as Java.

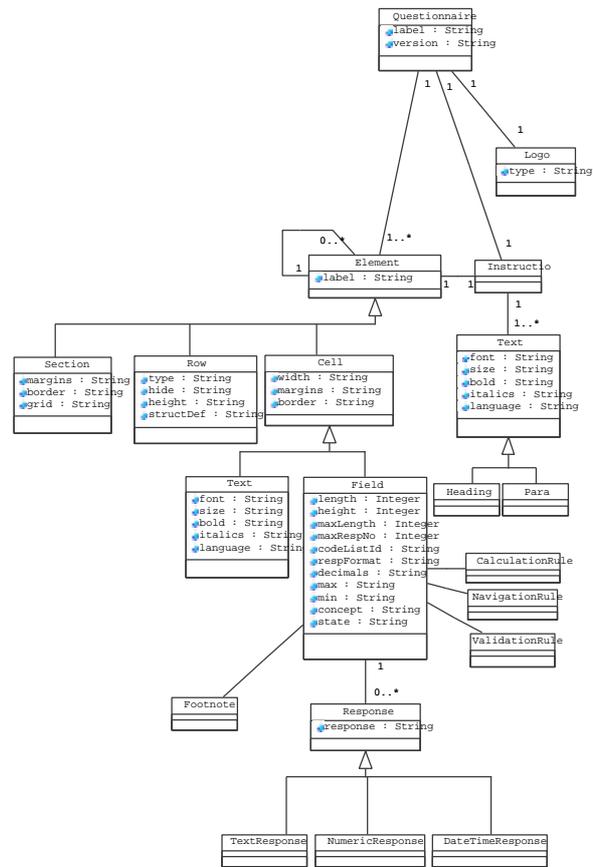


**Definition of a Structure**

The structure definition is central to the questionnaire object model, because this defines everything about the cells or tables on a form. Clearly, there are many such structures in any one questionnaire, as we shall see later.

This part of the object model has been used already to derive specifications for the exchange of multi dimensional data amongst statistical organisations and central banks. Indeed, the ongoing collection of data for the monitoring of the European economies by the European Central Bank, Eurostat (the statistical office of the European Commission), and the OECD has its roots in this model, for this is the model upon which the statistical exchange message GEMMES<sup>1</sup> is based.

However, as we shall see, this robust structure, is also appropriate to describe the structure of a questionnaire or other types of electronic form.



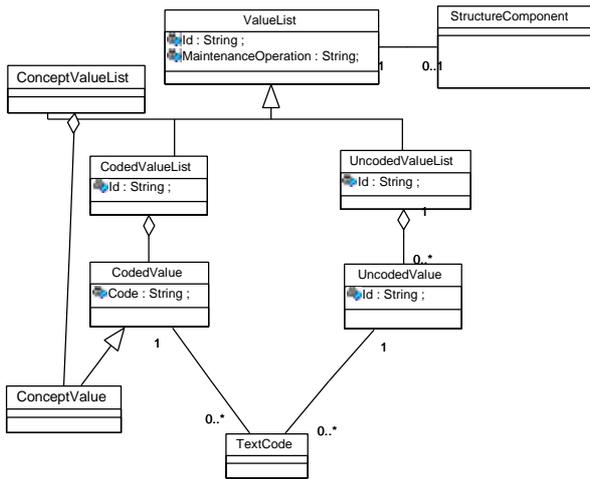
**The Questionnaire Object Model**

This model supports multiple hierarchies of elements, each of which can be a section, row, or column. Essentially, the section delineates one part of the questionnaire from another, the row is linked to the structure definition which defines precisely what the concepts or “data elements” are in the row (which can therefore be multi cell), and the code lists used and properties of each of the structure components or “columns”.

The Cell further defines the visualisation properties and the Response contains the data entered.

The only missing part of the model is the code list, which we have called a value list, as it could contain non coded items.

<sup>1</sup> GEMMES: Generic Statistical Message - an exchange message developed in the UN/EDIFACT syntax and also in XML



**The Value List Object Model**

### The XML DTD

The XML DTD can now be derived directly from the object model. There is no automated way of doing this, though some software has been tried (unsuccessfully). Each element in the DTD is the implementation of a class or an attribute in the object model. A part of the XML DTD for the Structure Definition is shown below.

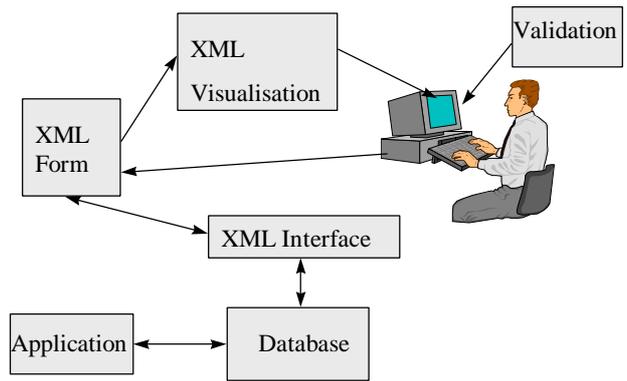
```

<!ELEMENT DataSetStructure
(StructureId, MaintenanceOperation?,
StructureComponent+) >
<!ELEMENT StructureComponent
(ConceptId, (TimeComponent |
DimensionComponent |
CellComponent |
AttributeComponent),
StructureComponentProperty*,
ValueListUsed?) >
<!ELEMENT StructureComponentProperty
(PropertyType, PropertyValue) >
<!ELEMENT PropertyValue (#PCDATA) >
<!ELEMENT PropertyType (#PCDATA) >
<!ELEMENT DimensionComponent
(Position) >
<!ELEMENT TimeComponent (Position)
>
<!ELEMENT AttachmentLevel (#PCDATA)
>
<!ELEMENT AttributeComponent
(AttachmentLevel?) >
<!ELEMENT ValueListUsed (#PCDATA)
>
<!ELEMENT CellComponent (Position)
>

```

### The System in Action

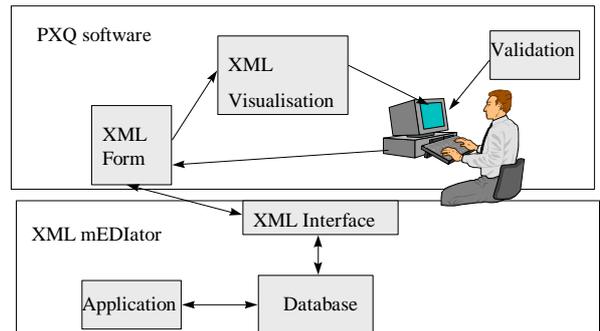
With the object model and the DTD defined it is now possible to write some implementations. One such implementation is described here.



**The XML e-form Scenario**

The XML Form can be created using a designer tool or simply by a text editor. It can be replicated and populated with pre-defined or known data from a database. It can be filled in on a screen using a plug-in to a browser, and the results written back to the XML document. The document can be downloaded from a website, filled in, re-named and sent back to the owner. The permutations are endless.

Two software products support this scenario.



**Software Support for XML E-forms**

### The PolyVal PXQ in Action

The following shows an XML-questionnaire after loading it in a browser. The example uses tick-boxes, drop lists, multiple choice and gives the possibility for a comment.

### Library Service Hours Form

In the XML-document it is possible to control both the contents and the appearance of the questionnaire with attributes such as width, height, type of field, etc.

```

<Section>
  <Row>
    <Cell width="40">
      <Text>Do you use any of our
        libraries?</Text>
    </Cell>
    <Cell width="30">
      <Field label="usage"
        codeList="YESNO">
        <Response></Response>
      </Field>
    </Cell>
  </Row>

```

The XML-application for questionnaires also gives the possibility of validating a response or to check one response against another, possibly resulting in an error message.

A calculation might be used to display a summary or to be used in a validation.

Navigation rules give the designer of a questionnaire the possibility to include rules on the order it is filled in, and also what is displayed and should be filled in.

All parts of the questionnaire that are language dependent have a language code associated with it, in order to display and operate in the selected language. This means that multiple language versions are embedded automatically in the one questionnaire.

## User Interface

As can be seen, both from the browser interface and the example XML section, some graphical user interface control has been included. A question that can be raised here is why XSL is not used. The view taken so far in this work is that XSL is not yet stable enough and that there are some doubts if it will be powerful enough. Another view is that in an application like this, we would loose control, being dependent on the browsers implementation of XSL.

However, when XSL has become stable enough it would be very easy to make use of it, as the structural aspects of the questionnaire are separated from the visualisation aspects in the DTD.

## Embedded browser application

The software that is used to view and respond to a questionnaire is developed as an embedded browser application and it is called the PolyVal XML Questionnaire (PXQ.EXE). This solution gives the possibility to load and respond to a questionnaire using HTTP on the Web. It also gives the possibility to run the software without a browser.

The PXQ software interprets the XML, so it can support any form that can be described using this XML.

## Advantages over HTML

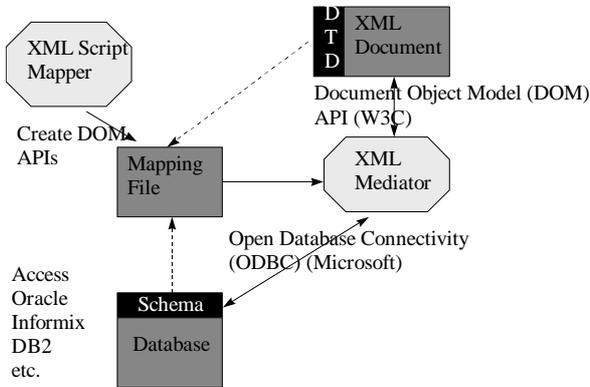
The XML approach has a number of clear advantages over using HTML and Java Script.

- It is scalable - no additional software development is required for additional forms- one software can support ALL forms
- Dynamic navigation is possible (answer Yes to box 3 and you next see box 7, or boxes 4-6 are skipped over and coloured grey). This can be done without the need to refresh the screen (as with HTML)
- The XML can be linked to a database easily, as the elements contain data type identifiers which can be processed directly by an application.
- The DTD will become a public standard and will be supported by free or inexpensive software

The PXQ-software also has well defined interfaces, in order for it to run together with other applications. Objects are defined for the different parts to be able to respond to a questionnaire from an application and a database. This also gives the respondent the possibility to set up and operate work flows within the organisation, making use of live objects.

## Linking the XML with a Database

As the XML is an open standard this can be supported by any software that complies to the XML DTD. One such software is the XML mEDIator.

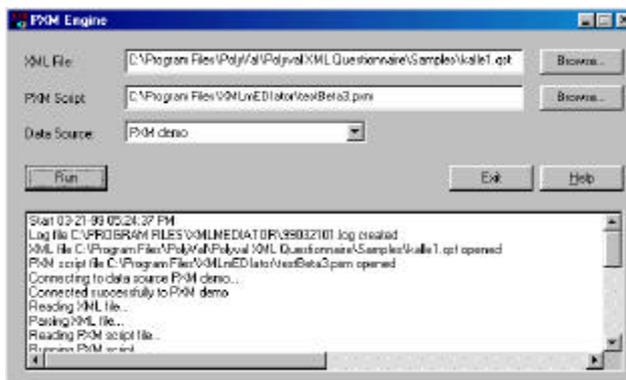


### The PolyVal XML mEDIator

This software uses a mapping file, created by a Script Designer, to map the XML document (DTD) to the database schema. The Script uses the Document Object Model (DOM) APIs to navigate the XML document, and SQL calls to read/write to the data base. The script is used at run time by the XML mEDIator and the relevant XML data is loaded into the database via ODBC calls.

## The XML mEDIator in Action

After an XML-document has been received it can be validated and loaded into a database with the PolyVal XML mEDIator. This software consists of two parts: one script designer and one engine to run a script against an XML-document. In the PXM Engine, the following must be selected: the XML-document to decode, the script to use, and the database to work with.



It is also possible to use the PXM Engine to read a database and to produce one or more XML-documents. This functionality is intended to be used to create pre-filled in questionnaires and for those that want to automatically create XML-documents from a database, such as Purchase Orders, Bookings, etc. The PXM Engine can also be used without a database: it can be used to create a CSV-file from an XML-document, convert one XML-document to another, etc.

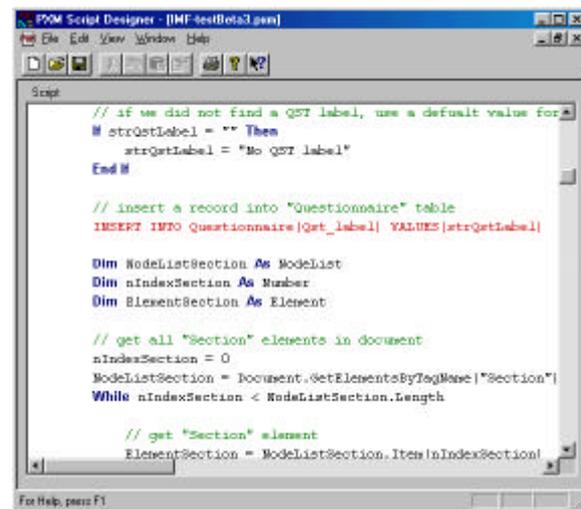
## Running the PXM Engine

The PXM Engine can use a script for its operations. This script contains DOM API's to navigate the XML-document and SQL-statements to operate against ODBC compliant databases. In the single user version shown above, an XML-file, a script and a database must be selected. When the PXM Engine executes it produces a log file. The server software does not require this selection, as this process is automated: the PXM Engine monitors a directory and depending on the type of XML document that is received, the appropriate script is selected.

It is also possible to run the PXM Engine from another software, using the defined objects.

## The Script Designer

The PXM Script Designer is used to create and maintain scripts. The script uses DOM and SQL statements in order to control the operations.



### PXM Script Design

The script is intended to be used when operating the PXM Engine without object interaction.

## Standards

It is important to standardise the XML DTD and its interfaces (IDL) in order to facilitate software interoperability. This will give the organisations a wide choice of software, which can be mixed and matched to suit the needs. It is not clear which standards organisation will be the custodian of XML DTDs, but it is important that such DTDs should be given a stamp of approval.

It is clearer which organisation will standardise the object interface. Although there are competing object standards, there is only one open standard, and this is CORBA which is managed by the Object Management Group (OMG). There is a group within the OMG that is standardising a number of XML DTDs for the interchange of metadata - this is called the XMI or Extensible Metadata Interchange standard. Much of the structure of the questionnaire model shown here can be termed metadata, for it describes the structure and code lists used by the questionnaire. It is the intention of the ADM Sig at the OMG to develop both native object standards (CORBA IDL) and XML specifications from the one object model. The extent to which the OMG will adopt the XML as part of their XMI work is not yet clear.

## References

[CORBA] Common Object Request Broker Architecture, Object Management Group, <http://www.omg.org>

[GESMES] Generic Statistical Message, Eurostat, [http://forum.europa.eu.int/dsis/Public\\_eeg6/library/dir/library/data/edi\\_pilot\\_projects/aggregated\\_gesmes?date](http://forum.europa.eu.int/dsis/Public_eeg6/library/dir/library/data/edi_pilot_projects/aggregated_gesmes?date)

[XMI] Extensible Metadata Interchange, Object Management Group, <http://omg.org/cgi-bin/doc?ad/98-07-01>

[XML] Extensible Markup Language, <http://www.w3.org>