

# xMod

## www.cch.kcl.ac.uk/xmod

### A TEI based publishing application

CCH  
Centre for Computing  
in the Humanities



KING'S  
College  
LONDON  
University of London

## Markup

We have written our own *in-house guidelines* for using TEI to create websites and by default, xMod relies on certain TEI markup conventions. However, it has been adapted to work on a variety of flavours of TEI and other XML vocabularies (including DocBook), and has even been used on occasions to publish data originally held in relational databases.

Where scholarly encoding is a key objective, each situation will inevitably have its own needs, but we have integrated this deeper markup with website presentational materials on a number of projects. It is impossible to predict every eventuality: what xMod does is to produce basic output options and at the same time allow you to carry out programming within a managed framework.

- **Examples of web publishing challenges:** multilingual websites, RSS feed mechanisms, sitemap features.
- **Examples of deeper encoding:** medieval charters, musicological bibliographies, classical inscriptions, biographies, glossaries, independent authority lists with extra information about people/places/key terms.
- **Integration with other web publishing tools:** integration of forms, image publication software (Zoomify), maps and plans.

## Management

Configuration of xMod is handled by a small set of key configuration files which offer extremely low level centralised control over the xMod process:

- **process:** sets key parameters, including information about the destination server.
- **filebase / navbar:** centralised management of individual pages and generation of navigation
- **imagebase:** centralised management of all image resources, including alternative text descriptions, captions, filetype, dimensions, multiple sizes.

Additional management of the process is possible via the extensive checking pages which are automatically generated when xMod is run. These allow XML encoders to view a comprehensive report of errors or missing data across the source XML repository.

## What is xMod?

xMod is a desktop application which can transform a repository of XML into a completely finished website.

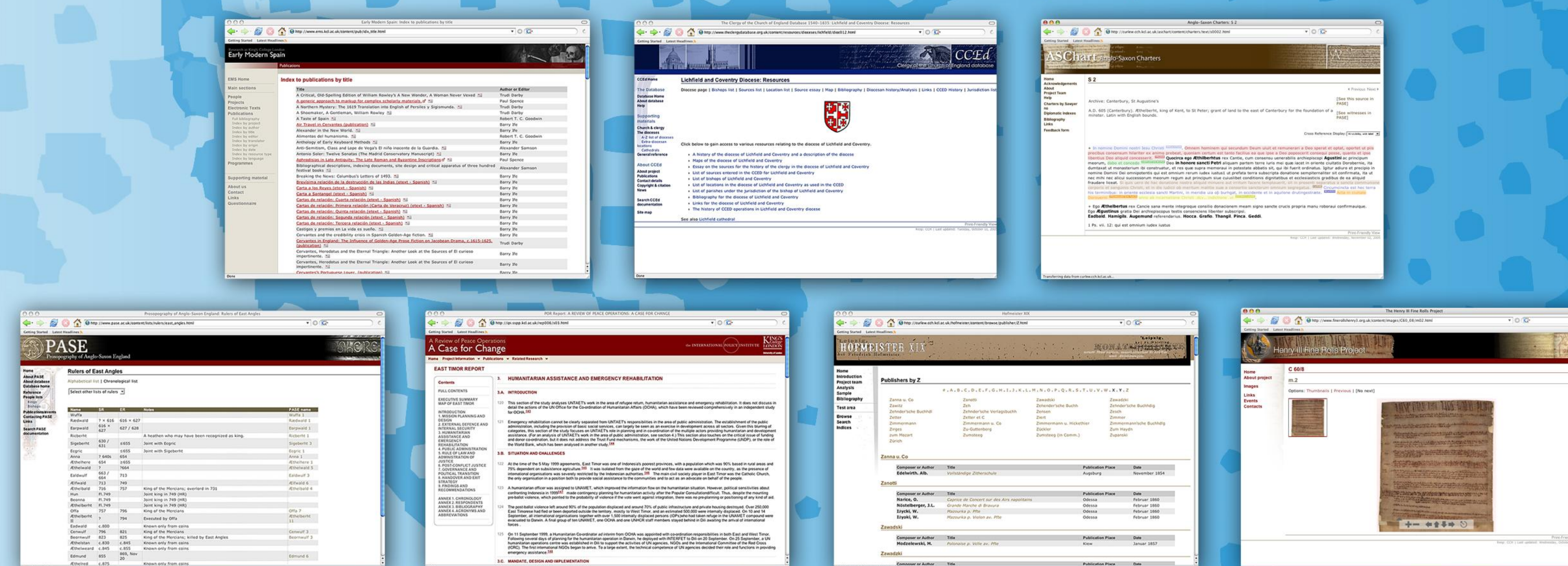
The entire process can be set up and run to produce a basic website very quickly, assuming some prerequisites:

1. **A set of valid XML files.** These would normally comply with a TEI DTD.
2. **Some basic configuration.** xMod needs to be told about the structure of the documents and the way in which they relate to one another (this forms the basis of the navigation system).
3. **A 'personality pack'.** xMod accepts a special set of CSS and image files which determine the look and feel. However, if they are not present, the completed website falls back on a default skin.

With this basic preparation done, the xMod application can be run and the resulting HTML output uploaded to a server.

*What makes it different?*

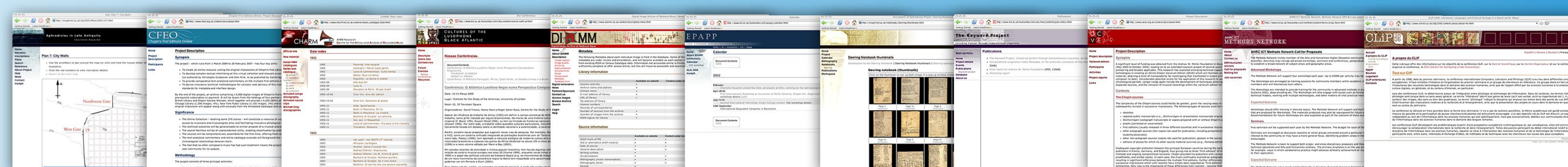
- xMod is not a component or tool for development, but an integrated package designed to output a usable, and potentially completely finished, website. Over the past two years, xMod has been used to generate a range of more than 30 vastly different websites.
- xMod is designed to support task-based workflow, with clearly defined and separate roles for an XML encoder, an XSL programmer and a Web Designer.
- xMod allows for extremely low level customisation, but this can be achieved without knowledge of how the application actually works.
- Alternatively, it is possible to run xMod without customisation and still produce a workable website.



## Future plans

The next milestone version will be 2.0, and much work has already been done towards rolling it out. It will add the following enhancements to xMod 1.x:

- Development branching into two versions:
  - Desktop version (2.0d) will switch to using Apache Ant for scripting to enable cross platform use (final desktop iteration);
  - Server version (2.0s) optimised to work with Apache Cocoon for dynamic website generation (trunk version);
- Addition of full text and semantic searching;
- Greater standards compliance: XSLT 2.0, XHTML 1.1;
- Increased separation of markup and presentation, with CSS-only (table-less) look and feel. Additionally a web application is being written to automate the process of generating and publishing the personality pack;
- Personality pack compatibility with another in house application used for managing database driven websites (rdb2java);
- Integration toolkit for real-time interoperability with our existing database web applications (AJAX connector).



## Core Logic

By default, xMod needs very little programming. It consists of a base layer of markup, publication/design options and supporting programming that covers the 70-80% of work common to all projects and provides a supporting framework for accommodating the 20-30% additional development work which makes each project unique.

- **Perl based core logic** which runs the process from beginning to end and performs file system operations
- **Batch file scripts** used to initialise the process and pass various parameters to the core logic
- Ability to **group files** for **generation of indices** at document and intra-document level
- Highly **modular XSLT structure**, with special libraries for different transformation scenarios
- The final page output is comprised of a **framework of content elements** (logo/banner, dual axis navigation) which can be made to change depending on context.

## Look and Feel

xMod has a generic look and feel, but is designed to use a small plug-in package of CSS, images and javascript called a **personality pack** to apply the look and feel to each website. This system is designed to support workflow by allowing a web/graphic designer to work on the look and feel independently of the rest of the development process. xMod makes comprehensive use of CSS, allowing for extremely low level customisation: no two sites need ever look the same. 'Personality packs' also include modular client side functionality where required, such as table sorting and accessibility tools.

In the current version of xMod (1.3) it is possible to alter the size of major content elements (navigation, banner area, table of contents, etc). xMod 2.0, currently in development, introduces the idea of **flexible content elements**, which will allow the designer to choose the onscreen position and size of each content element.