Supporting Medical Mathematics with Scriptable XML: The FORMOSA Language

Johannes W. Dietrich, Matthias F. Holzer, Martin R. Fischer

Medizinische Klinik, Campus Innenstadt, Klinikum of the University of Munich, F. R. Germany

and INSTRUCT AG Munich, F. R. Germany

Introduction

Mathematical methods have subtly gained a significant role in today's both theoretical and clinical medicine. Well established procedures in diagnostics and therapy planning rely on advanced numerical methods.

Nevertheless, the required calculations are not easily performed. According to several studies many physicians seem to have difficulties in performing calculations like determining drug doses or fluid balancing [Rolfe and Harper 1995, Potts and Phelan 1996, Lesar et al. 1997]. Apart from possible shortcomings in medical education these disenchanting results may predominantly be caused by poor availability of the required equations in the clinical setting and a lack of ease in performing the respective calculations.

Some years ago, our group developed a web based service (MFM, "Münchener Formelsammlung der Medizin / Munich Formulary of Medical Sciences") offering some essential equations along with an option to perform calculations for us with standard web browsers [Dietrich et al. 1997]. This CGI-based system was implemented as a combination of QTML (QuickTime Media Layer) and server-sided AppleScript. Although this prototype enjoyed an unexpected demand it was also faced with some intrinsic limitations. First, it was of course only accessible from computers connected to the web. In view of the target audience at smaller hospitals, general practitioners or in emergency medicine where internet access would come with regular charges this would prove as a relevant restriction. Furthermore, the AppleScript based calculation subsystem showed poor performance although running on a high performance RISC server. Finally, the data format proved to be not universal enough for all intended uses as HTML, specialized for web applications, only poorly separates formatting from content.

With these experiences we decided to reimplement the system nearly from scratch as a new program (FORMICA = formulary of mathematics in clinical application) providing the intended contents in several flavours: A stand alone application for common operating systems should be accomplished by an advanced web based service and a concise printed version with essential equations for clinical practice.

Therefore, we chose to develop a data format flexible enough to be easily converted to standard markup languages like HTML, XHTML or WML or to pre-press imaging standards as PostScript, PDF or MIF. Additionally, the format should be compact and robust and allow for fast rendering even on older machines.

Methods

FORMOSA, FORMICA's data format, is a language system consisting of an XML application (fML) and a scripting language (fScript). Unlike HTML with its plethora of detailed formatting tags fML focuses on content and therefore shows only a minimum of layout elements, predominantly to include multimedia objects like images or movies. To allow for reusing the calculation scripts of the MFM precursor project fScript was designed to resemble AppleScript, although in a simpler form.

A first version of the stand alone application has been developed for Macintosh computers with the THINK Pascal environment (Symantec Corporation, Cupertino, Ca., USA).

Following the demonstration of Niklas Wirth's PL/0 compiler [Wirth 1986] the parsing engine has been designed in a cascading manner beginning with a scanner routine reading in the respective next character from the FORMOSA source.

This scanner is used by a middle level service that, depending on the current context, either recognizes valid FORMOSA symbols or assigns them, should they be newly defined fScript denominators like variables or function definitions. This service again is used by higher level procedures that read whole XML elements, text blocks of encoded character data or fScript segments.

High level routines make use of these services to parse lists, entry blocks or forms written in fML or to interpret fScript code.

Data entry was realized with a FileMaker Pro Database (FileMaker Corporation, Santa Clara, Ca., USA) to allow editing calculation documents including multimedia objects and calculation scripts, before this information is read out via AppleScript to create XML files for inclusion as resources in the FORMICA application.

Results

A first beta version of the FORMICA stand alone application is available for Apple compatible Macintosh computers running under Mac OS 6.0.7 and newer, including classic mode in Mac OS X.

fML's content oriented vocabulary is sufficient to code informational screen cards, forms for calculations and content lists.

The hierarchical module-structure of the FORMOSA interpreter ensures a reliable parsing engine while keeping it transparent and compact. With a size of ca. 20 KB the parser is very small. Due to the content oriented XML syntax the FORMOSA documents are very compact, too: All documents showed a size of less than 50 % of the smallest HTML document showing the same functionality.

Card Editor in the Authoring Shell

FORMOSA Code

Equation Card in the FORMICA application
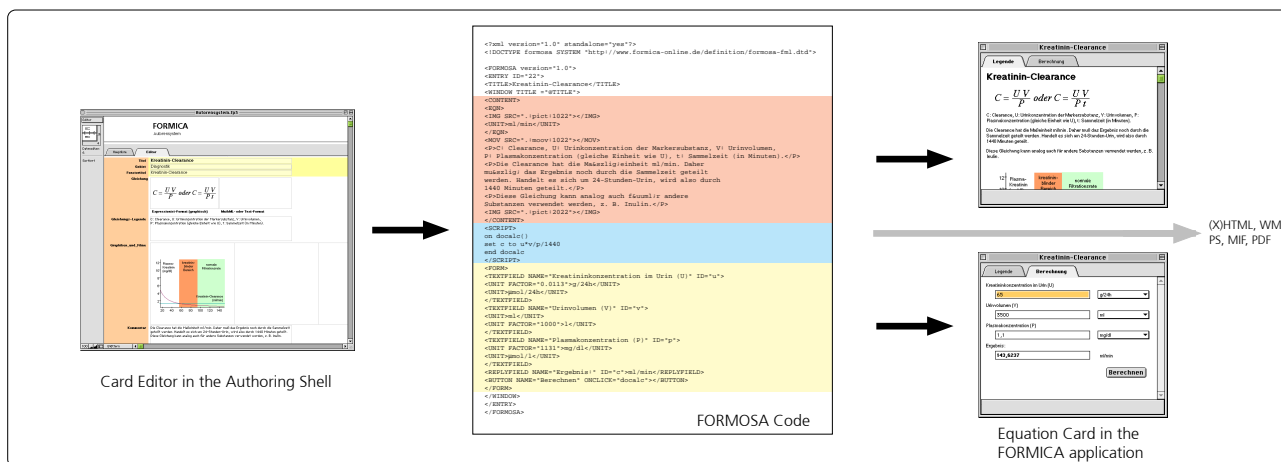
Outlook

The content-oriented approach of the FORMOSA language seems to be suitable for efficient coding of heterogeneous information containing multimedia elements and calculation instructions. The parser is pleasantly small, a fact caused by an only shallow implementation of error handling coming to less than 10 percent of the parser's code volume – a frugality that seems to be justifiable, as most FORMOSA code is automatically generated by the authoring shell.

In order to supply FORMICA for a broader audience the program is currently being ported to Windows (Delphi™ 5, Borland Corporation, Scotts Valley, Ca., USA) and to several Linux distributions (Debian, Mandrake and SuSe Linux with Borland Kylix™). With their advanced ObjectPascal implementation Delphi and Kylix ensure extensive source code compatibility to facilitate reusing of software components from the Macintosh platform. Porting to additional platforms like Mac OS X and Palm OS is currently under evaluation. These efforts are facilitated both by the platform independent format of the text based XML documents and by standard file formats for multimedia objects (PICT, QuickTime) that can be displayed either directly by the operating system (Mac OS) or by the freely available QuickTime extension (Windows) or appropriate open source substitutes (Linux).

Future developments will focus on creating a web based service that provides the contents for display with standard browsers. Depending on user agent information about the client browser this service will send either XML directly (linking to the FORMOSA DTD and a stylesheet document) or – for older browser versions – HTML/JavaScript that can efficiently be created from FORMOSA by simple "search and replace" operations. Other possible applications could cover automated conversion to standards like WML, PostScript or PDF to address an even broader audience.

FORMOSA's compactness shows that XML can be advantageously used even on older hardware while its transparency predestines it for flexible use where interactive web based applications are to be combined with the creation of a stand alone application and traditional print publishing.

http://www.formica-online.de

References

Dietrich, J. W., M. F. Holzer and M. R. Fischer (1997). "Die Münchener Formelsammlung der Medizin (MFM): Ziele und Implementation einer interaktiven Enzyklopädie im World Wide Web." Biomedical Journal 50 (Oct./Nov.): 8-10.

Lesar, T. S., L. Briceland, et al. (1997). "Factors Related to Errors in Medication Prescribing." JAMA 277 (No. 4): 312-7.

Potts, M. J. and K. W. Phelan (1996). "Deficiencies in Calculation and Applied Mathematics Skills in Pediatrics Among Primary Care Interns." Arch Pediatr Adolesc Med 150 (July 1996): 748-52.

Rolfe, S. and N. J. N. Harper (1995). "Ability of hospital doctors to calculate drug doses." BMJ 310 (6 May 1995): 1173-4.

Wirth, N. (1986). Compilerbau. Stuttgart, B. G. Teubner.