# Supporting Medical Mathematics with Scriptable XML: The FORMOSA Language

Johannes W. Dietrich, Matthias F. Holzer, Martin R. Fischer
Medizinische Klinik, Klinikum der Universität München, Campus Innenstadt, Munich (BY), F. R. Germany
and
INSTRUCT AG Munich (BY), F. R. Germany
eMail dietrich@medical-cybernetics.de

**Abstract**
In the past decades mathematical methods have gained a more and more important role for both theoretical and clinical medicine. Nevertheless, in the clinical setting they are only rarely applied, as the required equations and algorithms are not easily accessible.
To support clinical implementation even of advanced numeric methods we began to develop a new application with about 250 relevant equations based on experiences with an older web based information resource (Munich Formulary of Medical Sciences). Unlike the old system the new program should be made available in three forms: As a small stand-alone application on common computing platforms, as an internet site for standard browsers and as printed book. Therefore the data format should exhibit a structure that can be efficiently rendered even on older machines and allows for easy conversion to markup languages as HTML or WML as well as to imaging standards like PostScript or PDF.
Based on these requirements the decision fell on a language system (FORMOSA) composed of an XML application (fML) and a frugal scripting language (fScript) for processing of calculations.
The first version of the application was developed in Pascal for Mac OS computers. The parsing engine proved to be very fast and compact (< 50 KB) and therefore seems to be usable even on older machines as they are still in use in hospitals and at general practitioners.
As the XML structure is optimized for the contents the documents are very small, too (less than 50% of a comparable HTML file).
Currently the application is being ported to other operating systems (Windows and Linux with Delphi/Kylix). A further step will include a web service offering the contents in XML or HTML via automatical conversion, depending on the capabilities of the browser.

**KeyWords**
Cross-Platform Development, Medical Mathematics, Parsing, XML

## Introduction

Although still marginally perceived mathematical methods have gained a significant role in today's clinical medicine. Well established procedures as diagnostics of renal or pulmonary function or planning for radiation therapy rely on a plethora of numeric applications – apart from fields as epidemiology or medical decision theory that essentially serve as applied mathematics for health care.
Notwithstanding their decisive role the required calculations are not easily performed. Several studies imply that physicians often have difficulties to perform relevant calculations as determining drug doses or fluid balancing [Rolfe and Harper 1995, Potts and Phelan 1996, Lesar et al. 1997]. Apart from possible deficiencies in medical education these disenchanting results may predominantly be caused by poor availability of the required equations in the situation of clinical decision making and a lack of aid in performing the respective calculations.
In an earlier attempt to support clinical mathematics with technology our working group developed a web based service (MFM, "Münchener Formelsammlung der Medizin / Munich Formulary of Medical Sciences") that offered some essential equations along with an option to perform calculations for standard web browsers [Dietrich et al. 1997]. Technical basis for this CGI based system was a combination of QTML (QuickTime Media Layer) and server-sided AppleScript. The prototype enjoyed an unexpected demand, even outside the University of Munich, but it was also faced with some intrinsic limitations. First, it should of course remain unavailable to computers not connected to the web – in view of the usual hardware equipment at smaller hospitals, general practitioners or in emergeny medicine where internet access is often either impossible or comes with regular charges per connection this would prove as a relevant restriction. Furthermore, the AppleScript based calculation subsystem showed poor performance with response times of typically two to four seconds while

running on a fast RISC server – a tribute to the attempt for greater flexibility with server-scripts embedded in the QTML code [for details see Dietrich et al. 1997]. Nevertheless, the data format proved to be not universal enough for all intended uses as HTML, a critical component of the QTML convention, is specialized for web applications and thus separates formatting and content only poorly. Based on these experiences we decided to develop a new program (FORMICA = formulary of mathematics in clinical application) to offer the intended contents in several ways. A stand alone application for common operating systems should be accomplished by a web based service similar to the older system (while trying to avoid some of its shortcomings) and a printed version to provide a booklet with essential equations for clinical practice.

Consequently, 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, e. g. PostScript, PDF or MIF. Additionally, the format should be compact and robust and allow for fast rendering on older machines.

## Methods

FORMOSA, the data format for FORMICA, was designed as a duet of an XML application (fML) and a scripting language (fScript). Unlike HTML with its abundance of detailled formatting tags fML focusses on content and shows only a minimum of layout elements – predominantly to include multimedia objects like images or QuickTime-movies into the documents. To allow for re-using the calculation scripts of the MFM precurser project fScript was designed to resemble AppleScript, although it is structured simpler and focusses on mathematics. A formal definition of FORMICA's both braces is reprinted in the annex.

A first running version of the stand alone application has been developed for Macintosh computers with the THINK Pascal environment (version 4.5, Symantec Corporation, Cupertino, Ca., USA).

Following the demonstration of Niklas Wirth's frugal PL/0 compiler [Wirth 1986] the lowest level of the parsing engine has been constructed as a scanner routine (GetCh) that reads in the respective next character from the FORMOSA source (Fig. 1).

"GetCh" serves for a middle level routine (GetSym) that – depending on the current context – either recognizes valid FORMOSA symbols using a fast algorithm to detect the distance of strings or assigns them, should they be newly defined fScript denominators. "GetSym" again is used for a higher level procedure (GetTag) that reads whole tag structures and recognizes both symbols and attributes of XML tags. Other routines using GetCh read either whole text blocks or fScript segments for storing and parsing them if required.

High level routines make use of these services to parse lists, entry blocks or forms written in fML or to interpret fScript code (Fig. 1).

Data entry was realized with a FileMaker Pro Database that allows entering all elements of an equation document including multimedia objects and calculation scripts. In a second step these information is read out via AppleScript to create XML files that can be included as resources in the FORMICA application.

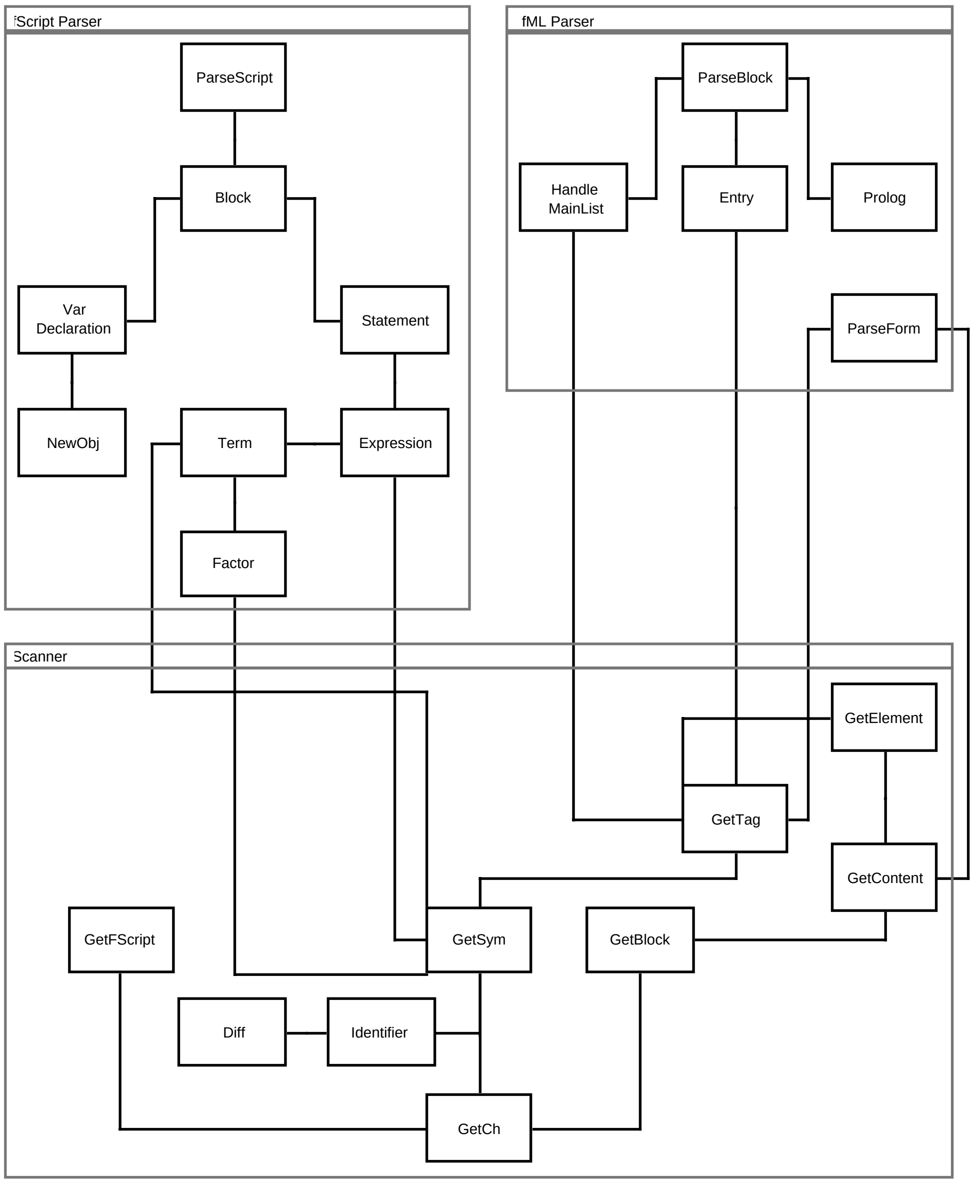


Fig. 1: Cascading module structure of the FORMOSA parsing engine

## Results

A first version of the FORMICA stand alone application is available in beta state for Apple compatible Macintosh computers (Fig. 2). It runs under Mac OS 6.0.7 through 9, including classic mode in Mac OS X.

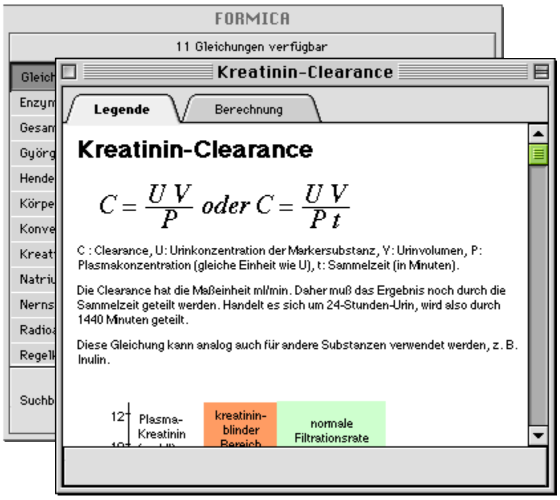


Fig. 2: Example of an equation document containing a legend and a calculation tab (see listing 1 for FORMOSA source code).

Listing 1: FORMOSA code for the equation document of Fig. 2:

```
<?xml version="1.0" standalone="yes"?>

<FORMOSA version="1.0">
<ENTRY ID="22">
<TITLE>Kreatinin-Clearance</TITLE>
<WINDOW TITLE ="@TITLE"></WINDOW>
<CONTENT>
<EQN>
<IMG SRC=".:pict:1022"></IMG>
<UNIT>ml/min</UNIT>
</EQN>
<MOV SRC=".:moov:1022"></MOV>
<P>C: Clearance, U: Urinkonzentration der Markersubstanz, V: Urinvolumen,
P: Plasmakonzentration (gleiche Einheit wie U), t: Sammelzeit (in Minuten).</P>
<P>Die Clearance hat die Ma&szlig;einheit ml/min. Daher
mu&szlig; das Ergebnis noch durch die Sammelzeit geteilt
werden. Handelt es sich um 24-Stunden-Urin, wird also durch
1440 Minuten geteilt.</P>
<P>Diese Gleichung kann analog auch f&uuml;r andere
Substanzen verwendet werden, z. B. Inulin.</P>
<IMG SRC=".:pict:2022"></IMG>
</CONTENT>
<SCRIPT>
on docalc()
set c to u*v/p/1440
end docalc
</SCRIPT>
<FORM>
<TEXTFIELD NAME="Kreatininkonzentration im Urin (U)" ID="u">
<UNIT FACTOR="0.0113">g/24h</UNIT>
<UNIT>µmol/24h</UNIT>
</TEXTFIELD>
```

```
<TEXTFIELD NAME="Urinvolumen (V)" ID="v">
<UNIT>ml</UNIT>
<UNIT FACTOR="1000">l</UNIT>
</TEXTFIELD>
<TEXTFIELD NAME="Plasmakonzentration (P)" ID="p">
<UNIT FACTOR="1131">mg/dl</UNIT>
<UNIT>µmol/l</UNIT>
</TEXTFIELD>
<REPLYFIELD NAME="Ergebnis:" ID="c">ml/min</REPLYFIELD>
<BUTTON NAME="Berechnen" ONCLICK="docalc"></BUTTON>
</FORM>
</ENTRY>
</FORMOSA>
```

fML's content oriented vocabulary is sufficient to code informational screen cards (see listing 1 and figure 2), forms for calculations and content lists (Fig. 3 and listing 2).


Fig. 3: Combined browsing and search window in FORMICA. The main list is built up from the FORMOSA code in linsting 2.

Listing 2: fML code for FORMICA's main list:

```
<?xml version="1.0" standalone="no"?>

<FORMOSA VERSION="1.0">
<!--Hauptliste-->
<MAINLIST>
<!-- Listenelemente, entry verweist auf den Eintrag -->
<CAPTION><COL1>Gleichung</COL1><COL2>Gebiet</COL2></CAPTION>
<ITEM ENTRY="4"><COL1>Enzymkinetik</COL1><COL2>Physiologie</COL2></ITEM>
<ITEM
ENTRY="2"><COL1>Gesamtk&ouml;rperwasser</COL1><COL2>Diagnostik</COL2></ITEM>
<ITEM ENTRY="5"><COL1>Györgi-Quotient</COL1><COL2>Diagnostik</COL2></ITEM>
<ITEM ENTRY="12"><COL1>Henderson-Hasselbalch-
Gleichung</COL1><COL2>Physiologische Chemie</COL2></ITEM>
<ITEM
ENTRY="52"><COL1>K&ouml;rperoberfl&auml;che</COL1><COL2>Diagnostik</COL2></ITEM>
<ITEM ENTRY="1"><COL1>Konversionskoeffizient</COL1><COL2>Physik</COL2></ITEM>
<ITEM ENTRY="22"><COL1>Kreatinin-Clearance</COL1><COL2>Diagnostik</COL2></ITEM>
<ITEM ENTRY="25"><COL1>Natrium-
Substitution</COL1><COL2>Therapieplanung</COL2></ITEM>
<ITEM ENTRY="52"><COL1>Nernstsche
```

```
Gleichung</COL1><COL2>Physiologie</COL2></ITEM>
<ITEM ENTRY="21"><COL1>Radioaktiver Zerfall</COL1><COL2>Physik</COL2></ITEM>
<ITEM ENTRY="9"><COL1>Regelkreis 0. Ordnung</COL1><COL2>Kybernetik</COL2></ITEM>
</MAINLIST>
</FORMOSA>
```

The hierarchical module-structure of the FORMOSA interpreter ensures a reliant parsing engine while keeping it transparent and compact (Fig. 1). With a size of 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. Parsing speed is comparable to rendering HTML documents with a modern HTML browser.

## Discussion and Outlook

The content oriented approach of the FORMOSA language seems to be suitable for efficient coding of heterogenous information containing multimedia elements and calculation instructions. All components allow for easy integration as resources into the viewer program itself, so that FORMICA acts as a self-contained application without requiring external files (and thus avoiding sources of error through possible inadequate user interventions).
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 for several Linux Distributions (Debian, Mandrake and SuSe Linux with Borland Kylix™). While offering a flexible ObjectPascal environment Delphi and Kylix ensure extensive source code compatibility to facilitate re-using of software components between the different platforms. Porting to additional platforms (Mac OS X and Palm OS) is currently under evaluation. These efforts are supported 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 free QuickTime extension (Windows) or appropriate open source substitutes (Linux).
The next step will create a web based service that provides the contents for display with standard browsers. Depending on user agent information this service will send either XML directly (linking to the FORMOSA DTD and a stylesheet document) or – for older browsers – HTML/JavaScript that can efficiently be created from FORMOSA by simple search and replace operations.
In the current development stage the parsing engine is not yet optimized for speed. In view of the fact that there are still several possible ways open to accelerate the system the current speed is satisfactory. FORMOSA's compactness shows that XML can be advantageously used even on older hardware while its transparency predestines it for flexible use in interactive web based applications as well as for traditional print publishing.

## Annex

*Formal Definition for FORMOSA*

```
<!-- DTD for FORMOSA (fML) 1.0 -->
<!-- (c) 1999-2001 J. W. Dietrich, University of Munich, Germany -->

<!ELEMENT formosa (mainlist | entry)>
<!ATTLIST formosa version CDATA #FIXED "1.0" #REQUIRED>

<!ELEMENT mainlist (caption, item+)>
<!ELEMENT caption (col1, col2)>
<!ELEMENT item (col1, col2)>
<!ATTLIST item entry ID>
<!ELEMENT col1 (#PCDATA)>
<!ELEMENT col2 (#PCDATA)>

<!ELEMENT entry (title, window.title?, (content | script? | form?)+)>
<!ATTLIST entry id ID>
<!ELEMENT title (#PCDATA)>
<!ELEMENT window.title ("@TITLE" | #PCDATA)>
```

```
<!ELEMENT content (eqn | img | mov | p | #PCDATA)*>
<!ELEMENT eqn (img, unit)>
<!ELEMENT img EMPTY>
<!ATTLIST img src CDATA #REQUIRED>
<!ELEMENT mov EMPTY>
<!ATTLIST mov src CDATA #REQUIRED>
<!ELEMENT p (#PCDATA)>

<!ELEMENT unit (#PCDATA)>
<!ATTLIST unit factor CDATA>

<!ELEMENT form ((textfield, unit)*, replyfield*, button)>
<!ELEMENT replyfield (#PCDATA)>
<!ATTLIST replyfield name CDATA>
<!ATTLIST replyfield id ID #REQUIRED>
<!ELEMENT button EMPTY>
<!ATTLIST button name CDATA>
<!ATTLIST button onclick CDATA>
<!ELEMENT textfield (#PCDATA, unit)>
<!ATTLIST textfield name CDATA>
<!ATTLIST textfield id ID #REQUIRED>

<!ELEMENT script ANY>
```

```
; EBNF Syntax for FORMOSA (fScript) 1.0
; (c) 1999-2001 J. W. Dietrich, University of Munich, Germany

script = block.

block = {"global" ident {"," ident}} {"on" ident "(" expression ")" block "end"
[ident]} statement.

statement = ["set" ident "to" expression | ident | "display dialog" expression
| "beep" [expression] | "if" condition "then" statement ["else" statement] "end
if" | statement].

condition = expression ("=" | "<>" | "<" | "<=" | ">" | ">=") expression.

expression = string | (["+" | "-"] term {("+" | "-") term}).

term = factor {("**" | "*" | "/") factor}.

factor = ident | number | "(" expression ")" | "sqr" factor | "sqrt" factor |
"sin" factor | "cos" factor | "exp" factor | "ln" factor | "arctan" factor.

ident = letter {letter | digit}.

number = digit {digit} ["." digit {digit}] ["e" ["+" | "-"] digit {digit}].

string = """ {character} """.

character = letter | digit.
```

**References**

Dietrich, J. W., M. F. Holzer und M. R. Fischer (1997). “Die Münchener Formelsammlung der Medizin (MFM): Ziele und Implementation einer interaktiven Enzyklopädie im World Wide Web.” *Biomedical Journal* 50 (Okt./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.

This work was supported by the Merck KGaA (EM Pharma), Darmstadt (HE), F. R. Germany