

6C-2

Emacs を利用した SGML エディタ

今郷 詔

(株)リコー 中央研究所

1 はじめに

一つの文書を紙への印字以外に、データベース作成や CRT 上での対話的な表示などにも使おうとすると、手続きを内包した文書表現ではなく、文書の構成要素とその間の関係が明示される表現が必要となる。このように文書構造を宣言的に記述する言語として SGML[1] が注目されている。

本稿では、まず SGML 文書入力の際の問題点を説明した後、テキスト・エディタの拡張として実現した SGML 支援機能とその実現方法を説明する。

2 SGML 文書入力の問題点

SGML 文書はテキスト・ファイルであるから一般のテキスト・エディタで作成することができる。しかし次のような制約があり、一般ユーザがこれらの条件を守りながらテキスト・エディタで文書を作成することは簡単ではない。

- 開始タグと終了タグを同じ element 内に対で入力しなければならない。
- 文脈によって使用可能なタグが変わる。
- すべてのタグの名前を正確に覚えておかねばならない。
- タグの名前をすべて入力しなければならない。
- 形式的な宣言が必要となる場合があるので、SGML の構文に関する知識が要求される。

これを解決するには SGML の構文を理解するエディタ [2] を使う必要がある。そのようなエディタは、構造エディタとして実現することもできるし、テキスト・エディタに SGML 支援機能を付加することで実現することもできる。

構造エディタとは SGML 文書の内部構造を直接編集するエディタで、文書の構文的な誤りは生じないために文書のデバッグが不要になる。しかし

SGML 入力専用となるため、まったく新しい操作体系を習得する必要がある。

一方、テキスト・エディタ方式では、SGML 以外の文書を作成するために使用しているエディタをそのまま使えるため、わずかな新しい操作を習得するだけでよく、柔軟な編集が可能である。しかし、文書の構文的な正しさは保証されないために文書のデバッグが必要となる。

SGML の知識があまりないユーザや、キーボードに不慣れなユーザには構造エディタが適するであろうが、キーボード入力に慣れているユーザにはテキスト・エディタ方式が適する。

本研究では、機能の拡張が容易なテキスト・エディタである GNU Emacs に SGML 編集機能を付加する方法を採用した。

3 SGML 支援機能

3.1 作成機能

SGML 文書の入力・編集を助けるための次のような機能がある。

- 文書の新規作成時には DTD 名を指定するだけで、必要な DOCTYPE 宣言を自動的に付加する。
- 非テキスト・メディア (イメージと音声) に対する入力インタフェースを持つ。
データの作成を行なうツールを呼び出し、作成したファイル名を名前とした entity 宣言を document type definition subset に付加すると共に、カーソル位置に entity 名を置く。例を図 1 に示す。
- 使用可能なタグの一覧をカーソル位置に応じて表示する。
ただし、element の数と順序に関する検査は行っていない。
- タグ名の補完を行なう。
部分的な名前からの入力から一意に決められる範囲で文字列を補う。

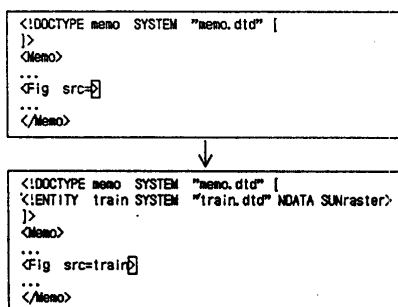


図 1: イメージ入力の場合

- 開始タグと終了タグを対にして挿入する。カーソルはその二つのタグの間におく。
- element の入れ子の深さに応じて自動的にインデントを行なう。

3.2 表示機能

テキスト以外のメディアとしてイメージと音声をサポートしている。イメージと音声は external entity 機構を使って文書とは別のファイルに収められており、その内容となるデータの符号化方式が entity 宣言に記されている。

カーソルを entity 名の上において、表示コマンドをタイプすると、その entity を内容に従って次のように表示または再生する。

音声 スピーカから再生する。

イメージ、テキスト 別ウィンドウに表示する。

ユーザは entity の中身を気にする必要はない。中身がイメージであれ音声であれ同じコマンドで表示や再生が行なえる。

4 エディタの実現方法

エディタの動作の大きな流れを図 2 に示す。

SGML 支援機能は Emacs lisp で記述され、Emacs の一つの main mode として実現されている。カーソル位置に対応する element の名前や、その element の入れ子レベルなどは、文書を文字列としてパターン・マッチングを行ない、開始タグと終了タグを認識することで得られる。

文書の新規作成時には、ユーザが指定した DTD 名から DTD の実体を引き出し、それをパーサが

解析してエディタが理解できる中間データを作成する。この中間データは Emacs lisp で記述されているため、そのまま Emacs にロードすることができる。

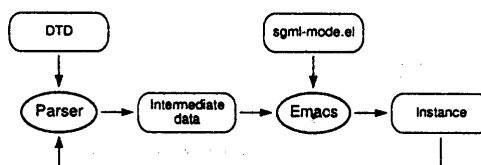


図 2: エディタの動作

中間データには次の三種類がある。

element 名リスト element 名の一覧で、タグ入力の補完時に使用する。

element 構造リスト カーソル位置での使用可能な element を表示する時に使用する。それぞれの element 名に対して、その子 element の名前をすべて記録したリストである。

このエディタでは element の数と順序に関する検査は行なわないので、occurrence indicator と connector に関する情報は持たない。

entity リスト entity によって参照している情報を表示する時に使用する。

entity 名に対して、その entity の内容・external entity であるかどうか・notation を記録したリストである。

5 おわりに

GNU Emacs に SGML 支援機能を付加することによって、わずかな新しい操作を覚えるだけで SGML 文書を容易に入力できるようになった。

今後は更に、開始タグにおける attribute 指定支援機能、使用可能 element 表示における occurrence indicator と connector 情報の利用ができるように拡張していく予定である。

参考文献

- [1] Charles F. Goldfarb. *The SGML Handbook*. Oxford University Press, New York, 1990.
- [2] ISO/IEC TR 10037. *Guidelines for SGML Syntax-Directed Editing Systems*, 1991.