Introduction pratique à SGML

Michel Goossens / CERN

Nanterre le 19 janvier 1995

- Pourqoui SGML ?
- Qui utilise SGML ?
- Les principes de base de SGML.
- La fonction et la structure d'une DTD.
- Quelques outils SGML.
- Autres normes pour les documents électroniques.

Pourquoi SGML ?

- passage support papier à support électronique ;
- grande masse d'informations structurées (annuaires, dictionnaires, recueils juridiqes) ;
- gestion des ces informations ;
- actualisation en permanence ;
- multiples représentations à partir d'un même document source (liste d'adresses, CD-ROM, minitel, WWW, base de données).

Points forts d'un balisage SGML

- une amélioration de la qualité des sources des documents ;
- une rationalisation dans le traitement du document, surtout un cycle de travail plus rapide ;
- une réduction du coût des publications ;
- la possibilité de réutilisation de l'information, d'où une plus-value (imprimées, hypertexte, bases de donnés).

La naissance de SGML

- une représentation de la structure logique des documents est essentielle pour les traiter électroniquement ;
- pour pouvoir échanger les documents il faut un langage commun ;
- en octobre 1986 ISO adopta officiellement SGML (*Standard Generalized Markup Language*), le langage standard de balisage généralisé ;
- adopté et utilisé par plusieurs organismes et entreprises nationaux et internationaux et par les développeurs de logiciels ;
- occupe déjà une place importante dans le monde de l'édition.

Qui utilise SGML ?

- l'association des éditeurs américains (AAP) (trois types de documents : le livre, la publication en série et l'article) ;
- l'AAP et l'EPS (European Physical Society) ont développé ISO 12083 (tableaux et formules mathématiques) ;
- CALS (*Computer-aided Acquisition and Logistic Support*) du département de la défense américaine (DoD) ;
- l'Office des Publications de la Communauté Européenne (FORMEX) ;
- l'association des éditeurs allemands (Börsenverein des Deutschen Buchhandels) ;
- la British Library avec « SGML : Guidelines for editors and publishers » et « SGML : Guidelines for authors» ;

Qui utilise SGML (suite) ?

- en France, le Syndicat national de l'édition et le Cercle de la librairie ont défini une application pour les éditeurs français ;
- l'Office des Publications de ISO (Genève) et le HMSO pour les brevets (Angleterre) ;
- Oxford University Press et Virginia Polytechnic (PhD, USA) ;
- le Text Encoding Initiative (textes classiques et commentaires) ;
- la plupart des manuels techniques, avec par ex. DTD `DocBook` ou autres, utilisés par IBM, HP, OSF, O'Reilly, etc.
- les modules SGML entrée/sortie sur des systèmes de traitement de texte ou de base de données (Frame, Interleaf, Microsoft, Oracle, Wordperfect) ;

Qui utilise SGML (suite) ?

- McGraw-Hill (Encyclopedia of Science and Technology) ;
- l'industrie électronique (Pinacle), l'industrie aéronautique, les compagnies aériennes (Boeing, Airbus, Rolls Royce, Lufthansa, etc.) et l'industrie pharmaceutique ;
- les agences de presse ;
- les éditeurs ou systèmes de présentation SGML (Arbortext, EBT, Exoterica, Grif, Softquad) ;
- HTML et WWW (évidemment !).

Principes de base de SGML

- une méthode normalisée pour représenter l'information contenue dans un document ;
- indépendant des systèmes de saisie et de traitement ;
- indépendant de la forme physique finale ;
- principe de *balisage* logique généralisé ;
- ne définit pas le langage de balisage ;
- un *méta-langage* pour construire plusieurs langages de balisage.

Le balisage spécifique

TEX

```
\vfil\eject
\par\noindent
{\bf Chapitre 2 : Titre du chapitre}
\par\vskip\baselineskip
```

Script

```
.pa
.bd Chapitre 2 : Titre du chapitre
.sp
```

Le balisage spécifique (suite)

- balisage de type « présentation » : caractères de contrôle mêlés aux caractères (imprimables) du document. (fin de ligne, coupures de page, contrôle d'espacements...) ;
- ces caractères sont non-standard ;
- difficile d'échanger des documents entre différents systèmes ;
- balisage propre à une représentation donnée et une présentation physique ;
- facile à afficher et imprimer ;
- n'autorise que des traitements limités.

Balises génériques ou logiques

LaTeX

```
\chapter{Titre du chapitre}
\par
```

HTML (SGML)

```
<H1>Titre du chapitre</H1>
<P>
```

Balises génériques ou logiques (suite)

- augmenter les possibilités de traitement ;
- faire abstraction de tout aspect physique ;
- décrire la fonction logique des éléments (titre, sections, paragraphes, tableaux, références bibliographiques…) ;
- spécifier leurs relations mutuels.

Balisage logique généralisé

Le *marquage* de la structure d'un document, s'effectue en deux temps :

1. la définition de l'ensemble des balises pour identifier les éléments d'un document et les règles formelles qui décrivent sa structure (c'est le rôle de la DTD) ;
2. l'introduction du balisage dans le document lui-même, selon les principes de cette définition formelle.

```
<article>
<tit>Une introduction à SGML</tit>
<sec>Principes de base de SGML</sec>
<P>  ...
<ssec>Le balisage logique généralisé</ssec>
<P>  ...
```

Balisage logique généralisé (suite)

Plusieurs documents spécifiques peuvent appartenir à une seule classe de documents construits selon la même structure générique.

```
     Article A                    Article B
Titre                        Titre
  Section 1                    Section 1
    Sous-section 1.1             Sous-section 1.1
    Sous-section 1.2             Sous-section 1.2
  Section 2                        Sous-section 1.3
  Section 3                    Section 2
    Sous-section 3.1             Sous-section 2.1
    Sous-section 3.2             Sous-section 2.2
    Sous-section 3.3         Bibliographie
    Sous-section 3.4
Bibliographie
```

La définition d'une classe de document (DTD)

La DTD définit :

- le *nom* des éléments qui peuvent être utilisés ;
- le *contenu* de chaque élément ;
- *combien de fois* et dans quel *ordre* les éléments peuvent apparaître ;
- si une balise de début (fin) peut être *omise* ;
- les *attributs* éventuels et leur valeurs par défaut ;
- le nom des *entités* qui peuvent être utilisées.

La structure de la DTD

La DTD de HTML2 contient deux parties :

1. partie « système » : le jeu de caractères utilisé et les options autorisées, par ex. `OMITTAG` ;
2. partie « déclaration » de la classe de document `HTML` : les éléments, attributs et entités possibles.

Les déclarations ont la forme :

```
<! ... >
```

Une section marquée a la forme :

```
<![ mot_clé [ ... ]] >
```

La structure de la DTD (suite)

Les commentaires

```
<!-- texte du commentaire -->
```

La déclaration d'un élément

- nom de l'élément ;
- la minimisation possible ;
- le *modèle de contenu*.

```
<!ELEMENT (OL|UL) - -  (LI)+>
<!ELEMENT LI      - O  %flow>
```

La structure de la DTD (suite)

Symboles d'ordre et de choix

,	« et » ordonné ;	+	1 fois ou plus ;
&	« et » non ordonné ;	?	0 ou 1 fois ;
|	« ou » exclusif ;	*	0 fois ou plus.

```
<!ELEMENT DL    - -  (DT*, DD?)+>

<!ENTITY % head.content "TITLE & ISINDEX? & BASE? & META*
                         %head.nextid %head.link">
<!ELEMENT HEAD O O  (%head.content)>
<!ELEMENT FORM - - %body.content
          -(FORM) +(INPUT|SELECT|TEXTAREA)>
```

La structure de la DTD (suite)

Type de caractères utilisés

`PCDATA` *parsed character data* ou données textuelles analysées.

```
<!ELEMENT TITLE - - (#PCDATA)>
```

`RCDATA` *replaceable character data* ou données textuelles remplaçables.

`CDATA` *character data* ou données textuelles.

```
<!ELEMENT TEL CDATA>
```

`ANY` L'élément peut contenir du matériel de type `PCDATA` ou tout autre élément défini dans la DTD.

`EMPTY` L'élément a un *contenu vide*.

```
<!ELEMENT IMG - O EMPTY>
```

La structure de la DTD (suite)

Les attributs

Une déclaration d'attribut comporte :

- le nom de l' (des) élément(s) auquel elle se rapporte ;
- le nom de l'attribut ;
- soit le *type de l'attribut*, indiqué par un mot-clé soit, entre parenthèses, la liste des valeurs que peut prendre cet attribut ;
- une valeur par défaut (sous la forme d'une des valeurs autorisées, spécifiée entre guillemets, ou d'un mot-clé).

```
<!ATTLIST élément_qualifié attribut (valeurs) "défaut">
```

La structure de la DTD (suite)

Les mots-clé pour les types d'attribut

`CDATA`	données textuelles (caractères quelconques) ;
`ENTITY(IES)`	nom(s) d'entité(s) générale(s) ;
`ID`	l'identificateur SGML d'un élément ;
`IDREF(S)`	valeur(s) d'appel d'identificateur(s) d'élément ;
`NAME(S)`	nom(s) SGML ;
`NMTOKEN(S)`	unité(s) lexicale(s) nominale(s) ;
`NOTATION`	nom de notation ;
`NUMBER(S)`	nombre(s) ;
`NUTOKEN(S)`	unité(s) lexicale(s) numérique(s).

La structure de la DTD (suite)

Les mots-clé pour les valeurs par défaut

`#FIXED`	L'attribut a une valeur fixe, et ne peut prendre que cette valeur ;
`#REQUIRED`	Une valeur doit obligatoirement être spécifiée par l'utilisateur ;
`#CURRENT`	Si une valeur n'est spécifiée, la valeur par défaut utilisée sera la dernière valeur spécifiée ;
`#CONREF`	La valeur sera utilisée pour les références croisées ;
`#IMPLIED`	Si une valeur n'est spécifiée, le système de traitement définira une valeur.

La structure de la DTD (suite)

```
<!ATTLIST DL COMPACT (COMPACT) #IMPLIED>

<!ATTLIST PRE WIDTH NUMBER #implied>

<!ATTLIST IMG SRC   %URI;                #REQUIRED
              ALT   CDATA                #IMPLIED
              ALIGN (top|middle|bottom)  #IMPLIED
              ISMAP (ISMAP)              #IMPLIED >
```

Les entités

Des entités sont utiles dans plusieurs circonstances :

– notations raccourcies pour des suites de caractères saisies fréquemment (entités générales) :

```
<!ENTITY GUT "GUTenberg">
```

– notations pour saisir des caractères spéciaux, accents ou symboles (entités générales et caractères).

```
<!ENTITY amp CDATA "&#38;" -- << et >> commercial "&"-->
```

L'ISO a défini plusieurs ensembles d'entités caractères standardisés, les symbôles graphiques, mathématiques, etc.

– l'inclusion de fichiers externes (entités externes) ;

```
<!ENTITY article SYSTEM "/usr/g/goossens/sgmlart.sgml" >
```

Les entités (suite)

Il faut saisir les entités textuellement en faisant attention à la présence de majuscules ou minuscules. La casse du nom des éléments et des attributs n'a pas d'importance.

appels d'entités

```
&GUT;
```

Les entités paramètres

Utilisées à l'intérieur d'une DTD pour augmenter la modularité de la définition des différents éléments de la DTD.

```
<!ENTITY % heading "H1|H2|H3|H4|H5|H6">
<!ENTITY % list " UL | OL | DIR | MENU " >
<!ENTITY % text "#PCDATA | A | IMG | BR">

<!ELEMENT ( %heading )  - -  (%text;)+>
```

Quelques outils SGML

- SGML est maintenant très répandu ;
- de multiples solutions commerciales existent ;
- augmenter la productivité, le confort et la convivialité d'utilisation ;
- quelques outils intéressants et disponibles publiquement.

Valider un document SGML avec `sgmls`

– `sgmls` est un programme d'analyse (*parser*) disponible publiquement ;

– développé par James Clark en se basant sur un programme antérieur `arcsgml` Charles Goldfarb ;

```
sgmls  [-deglprsuv] [-cfichier] [-inom] [nom_de_ficher]
```

Valider un document SGML avec `sgmls` (suite)

```
<HTML>
<!-- Un commentaire -->
<HEAD>
  <TITLE>Document test HTML</TITLE>
</HEAD>
<!-- Début du corps du document -->
<BODY>
<DL>
  <DT>terme 1<DD>donnée 1
  <DT>terme 2<DD>donnée 2
  <DT>terme 3
  <DT>terme 4<DD>donnée 4<DD>donnée 4 bis
</DL>
</BODY>
</HTML>
```

Valider un document SGML avec `sgmls` (suite)

```
(HTML
(HEAD
(TITLE
-Document test HTML
)TITLE
)HEAD
(BODY
ACOMPACT IMPLIED
(DL
(DT
-terme 1
)DT
(DD
-donnée 1\n
)DD
(DT
-terme 2
)DT
(DD
-donnée 2\n
)DD
(DT
-terme 3\n
)DT
(DT
-terme 4
)DT
(DD
-donnée 4
)DD
(DD
-donnée 4 bis
)DD
)DL
)BODY
)HTML
```

Valider un document SGML avec `sgmls` (suite)

Un document contenant une erreur :

```
<HTML>                                  ligne 453
<BODY>                                        454
  <P>texte dans un paragraphe                 455
</BODY>                                       456
</HTML>                                       457
```

Valider un document SGML avec `sgmls` (suite)

```
AVERSION CDATA -//IETF//DTD HTML//EN//2.0
(HTML
(HEAD
sgmls: SGML error at a, line 454 at ">":
       BODY element not allowed at this point in HEAD element
(P
-texte dans un paragraphe
)P
sgmls: SGML error at a, line 457 at ">":
       HEAD element ended prematurely; required subelement omitted
)HEAD
sgmls: SGML error at a, line 457 at ">":
       HTML element ended prematurely; required BODY omitted
)HTML
```

Valider un document SGML avec `sgmls` (suite)

Un autre document contenant une erreur :

```
<HTML>                                  ligne 453
<HEAD>                                        454
<TITLE>titre</TITLE>                          455
</HEAD>                                       456
<BODY>                                        457
<LI>                                          458
</BODY>                                       459
</HTML>
```

Valider un document SGML avec `sgmls` (suite)

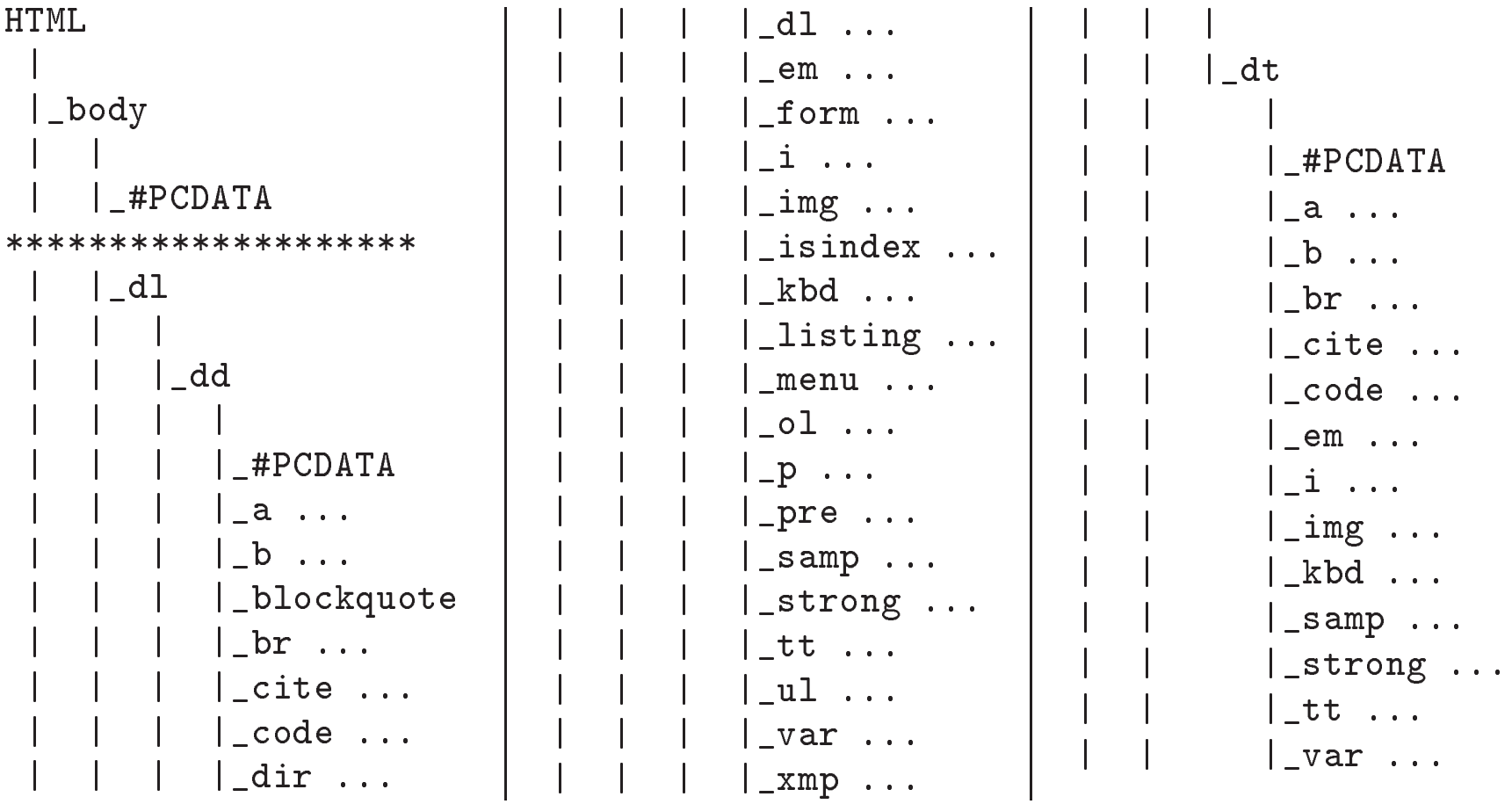
```
(HTML
(HEAD
(TITLE
-titre
)TITLE
)HEAD
(BODY
)BODY
sgmls: SGML error at a, line 458 at ">":
       Out-of-context LI start-tag ended HTML document element
                                                    (and parse)
)HTML
```

Outils d'analyse de documents SGML

Earl Hook a développé plusieurs outils SGML écrits en `perl`, permettant d'analyser un document ou DTD SGML :

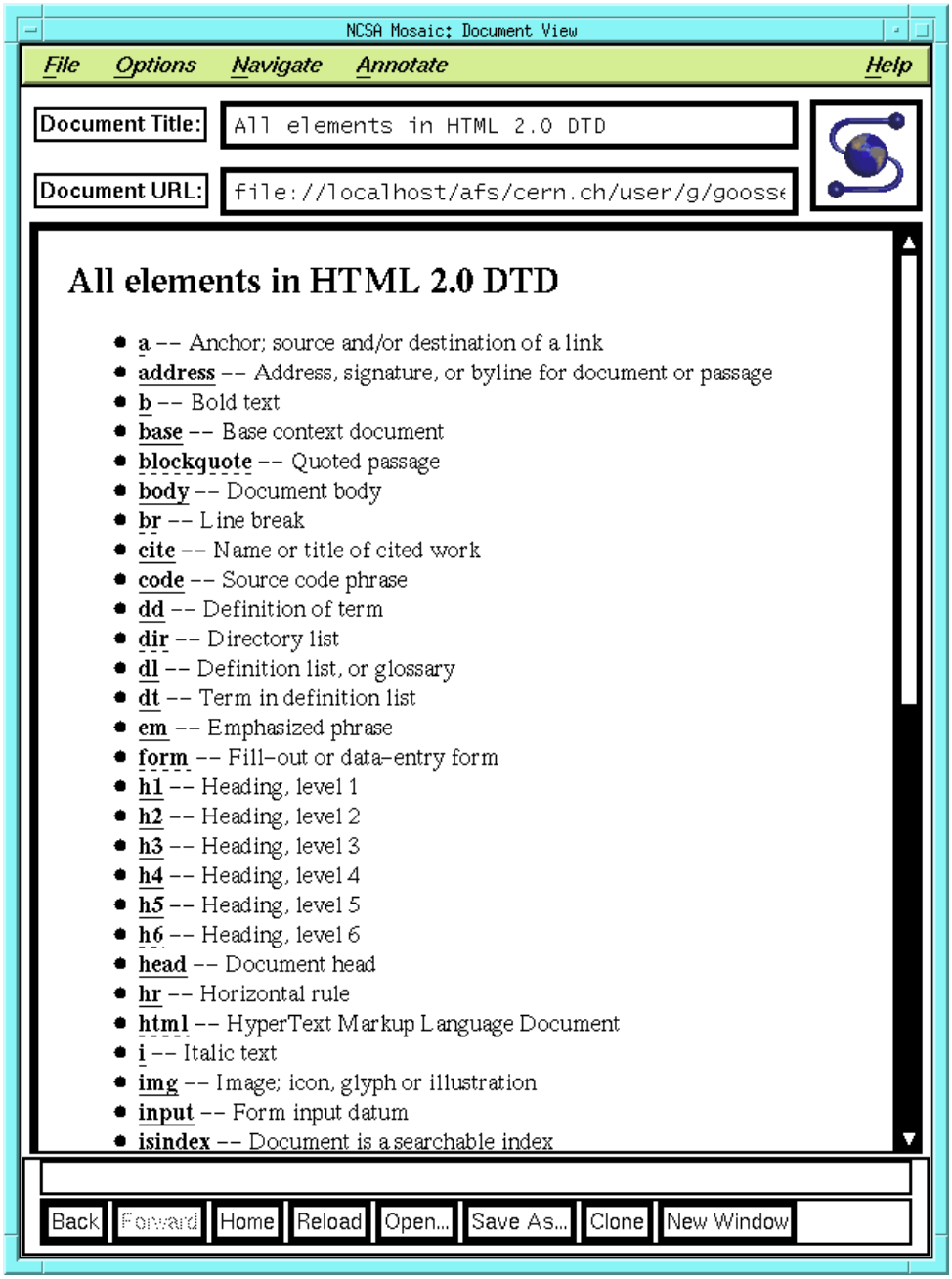
`dtd2html`	produit un document HTML à partir d'une DTD SGML qui permet une navigation hypertexte à travers une DTD SGML ;
`dtddiff`	compare deux DTDs et montre les différences éventuelles ;
`dtdtree`	produit une visualisation de l'arborescence hiérarchique caractérisant les relations entre les différents éléments définis dans une DTD ;
`stripsgml`	enlève les balises SGML d'un texte et essaie de traduire les appels d'entités de caractères standard en ASCII.

Étudier la structure d'une DTD — dtdtree



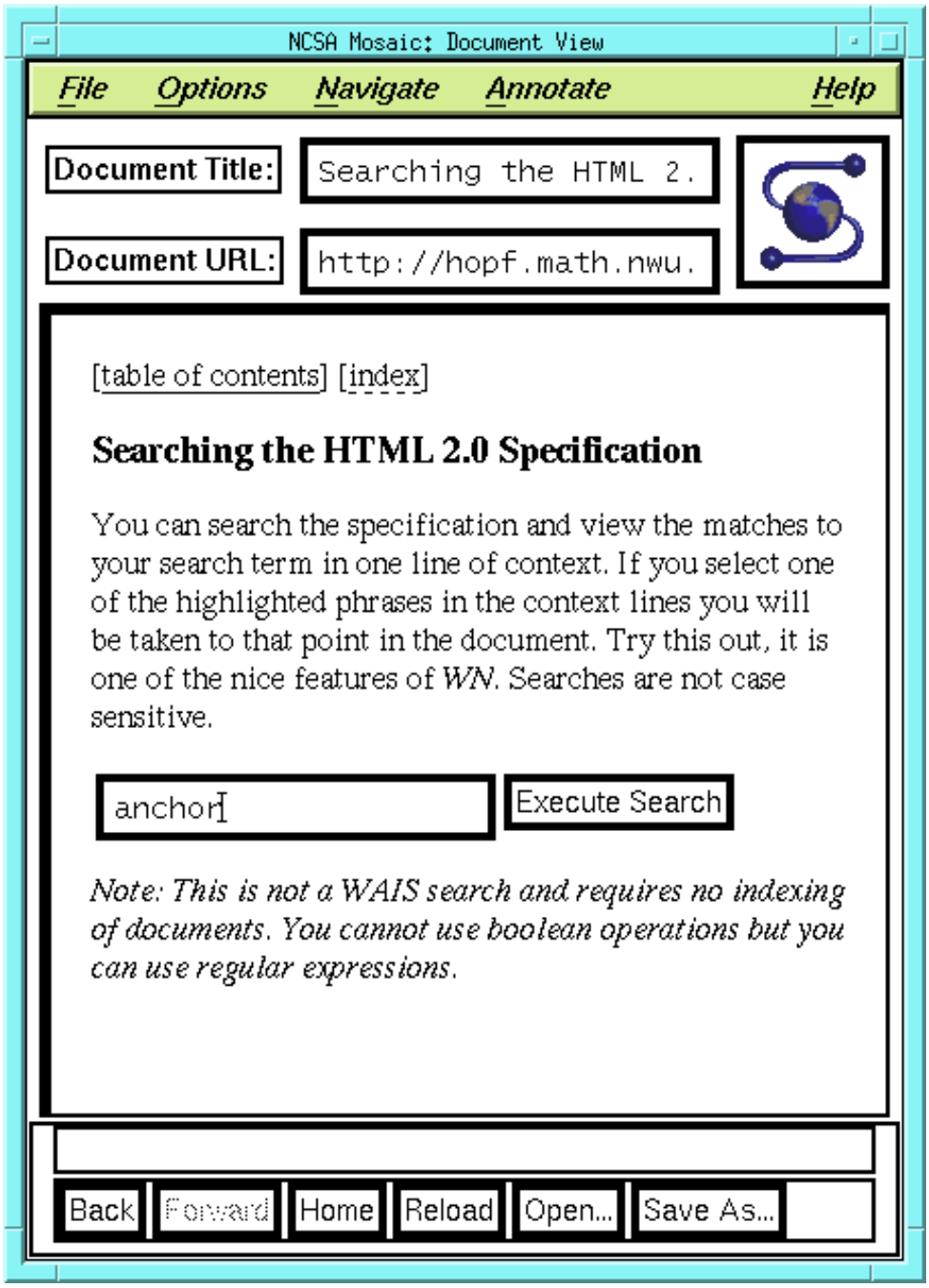
Documenter une DTD – `dtd2html`

Description hypertexte des éléments d'une DTD (HTML2) visualisée avec `mosaic`.



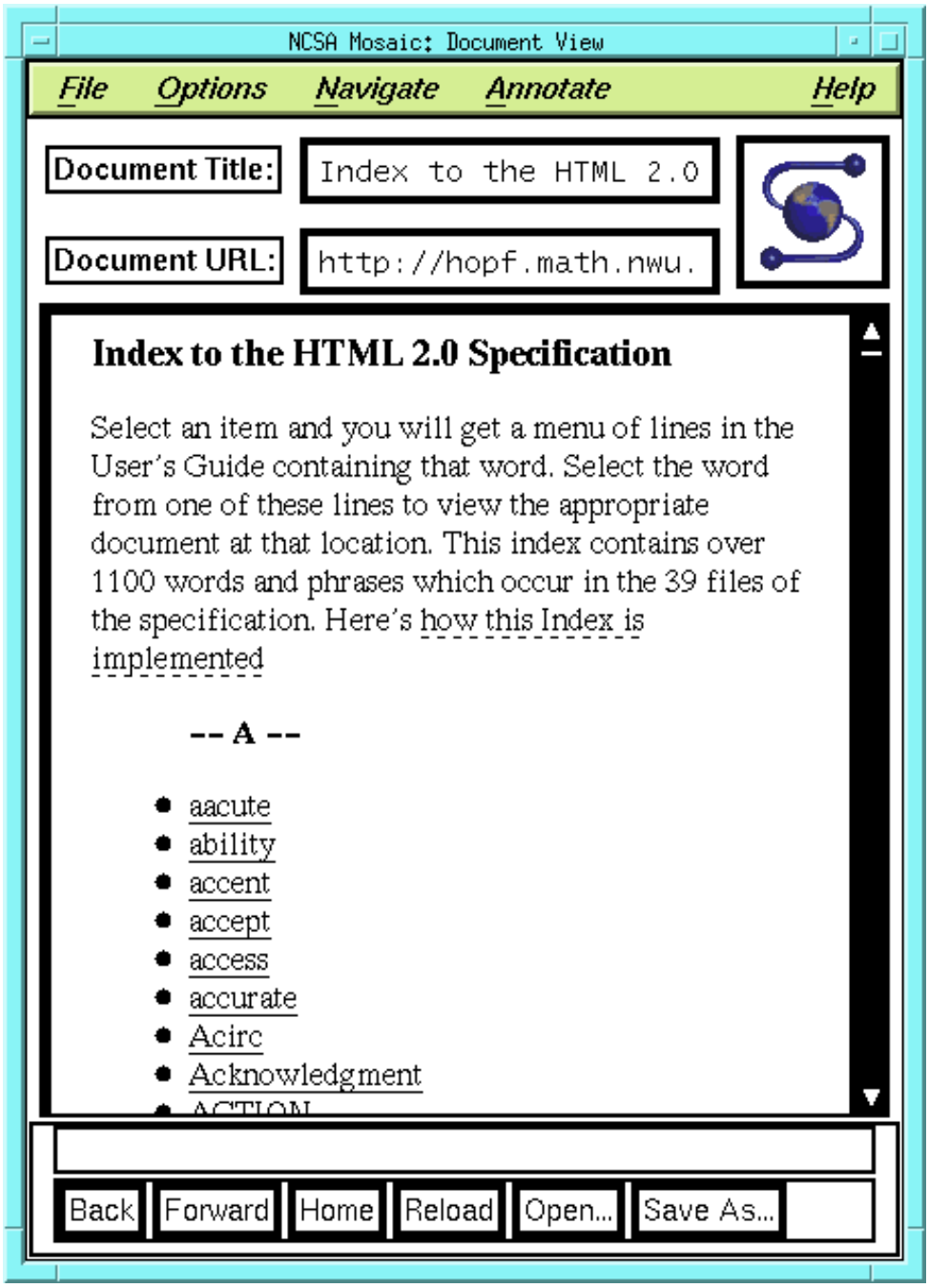
Recherche d'information

Un moteur de recherche par expression régulière.



Recherche en index

Index de 1100 mots et phrases de la DTD HTML2.



Autres normes

- ISO 8879 le standard SGML ;
- ISO 9096 (SDIF) format pour échanger des documents SGML ;
- ISO 10744 (Hytime) formalisme pour la représentation hypermedia de documents.
- ISO-DIS 10179 (DSSSL pour *Document Style Semantics and Specification Language*) les concepts et actions nécessaires pour passer de la structure logique d'un document à sa mise en forme physique.
- ISO-DIS 10180 (SPDL pour *Standard Page Description Language*) description des documents dans leur forme finale, entièrement composée, non révisable (presque du PostScript) ;
- ISO 9541 le standard des polices de caractères. Méthode pour nommer et grouper les glyphes ou collections de glyphes indépendante de tout codage spécifique.