UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Feature grammar systems. Incremental maintenance of indexes to digital media warehouses

Windhouwer, M.A.

Publication date
2003

Link to publication

Citation for published version (APA):
Windhouwer, M. A. (2003). *Feature grammar systems. Incremental maintenance of indexes to digital media warehouses*. [Thesis, externally prepared, Universiteit van Amsterdam].

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:18 Nov 2026

Appendix B

Feature Grammars

B.1 The WWW Feature Grammar

```
%module     WWW;

%start      WebObject(Location);

%detector   WebHeader(Location);
%detector   Robot(Location,"AcoiRobot");
%detector   Explored(MIME);
%detector   Tried(parent::Status);

%atom       www::url {(^http://([^ :/]*)(:[0-9]*)?/?(.*)$)
                     |(^file://(.*)$)};
%atom       temporal::date;

%atom       url  Location;
%atom       date Modification;
%atom       lng  Length;

WebObject   : Location (Robot WebHeader WebBody)? Status;
WebHeader   : Redirect? MIME Modification? Length?;
MIME        : Primary Secondary;
Status      : Explored | Tried;
```

B.2 The Text Feature Grammar

```
%module     Text;

%use        WWW;
```

```
3  %detector  TextType [ MIME/Primary = "text" ];

4  %detector  DRUID(Location);

5  %atom str  Language;

6  WebBody    : TextType Text;
7  Text       : DRUID;
8  DRUID      : Language;
```

B.3 The HTML Feature Grammar

```
1  %module    HTML;

2  %use       Text;

3  %detector  HTMLType [ MIME/Secondary = "html" ];
4  %detector  HTML(Location);

5  %atom str  Title, Word, Link, Alt;
6  %atom bit  Embedded;

7  WebBody    : HTMLType HTML;
8  HTML       : Title Body? Anchor*;
9  Body       : &Keyword+;
10 Anchor     : &WebObject Embedded Link? Alt?;

11 %start     Keyword(Word);

12 %detector  Synset(Word);

13 %start     Synonyms(Word);
14 %start     Hypernyms(Word);
15 %start     Hyponyms(Word);

16 Keyword    : Word Synset;
17 Synset     : &Synonyms &Hypernyms &Hyponyms;
18 Synonyms   : id str+;
19 Hypernyms  : id str+;
20 Hyponyms   : id str+;
```

B.4 The Image Feature Grammar

```
%module    Image;

%use       WWW;

%detector  ImageType [ MIME/Primary = "image" ];
%detector  Portrait [ Faces/Number = 1 ];

%detector  Global(Location);
%detector  Icon(Location);
%classifier decrules::Photo(Global);
%classifier decrules::Graphic(Global);
%detector  Skin(Location);
%detector  Faces(Location);
%detector  exec::Histogram(Location);

%atom vector::flts;

%atom int  Number;
%atom flt  Prevalent, Far, NormalizedFar, Saturation, Percentage;
%atom flts HSB, RGB;
%atom bit  Animated;

WebBody    : ImageType Image;
Image      : Global Icon Class;
Global     : Size Color Animated;
Size       : Width Height Ratio;
Color      : Number Prevalent Neighbor Saturation Histogram;
Neighbor   : Far NormalizedFar;
Histogram  : RGB HSB;
Icon       : Location;
Class      : Graphic | Photo (Skin Faces Portrait?)?;
Skin       : Percentage;
Faces      : Number;
```

B.5 The Audio Feature Grammar

```
%module    Audio;

%use       WWW

%detector  AudioType [ MIME/Primary = "audio" ];

WebBody    : AudioType Audio;
```

B.6 The MIDI Feature Grammar

```
1  %module    MIDI;

2  %use       Audio;

3  %detector  MIDIType [ MIME/Secondary = "midi" ];

4  %detector  exec::MIDI(Location);

5  %atom int  QuarterNode, Id, Channel, Track;
6  %atom str  Lyrics, Name, Contour;

7  Audio      : MIDIType MIDI;
8  MIDI       : QuarterNode Musician* Lyrics* Profile*;
9  Musician   : Instrument Channel;
10 Instrument : Id Name?;
11 Profile    : Track Contour;
```

B.7 The MP3 Feature Grammar

```
1  %module    MP3;

2  %use       Audio;

3  %detector  MP3Type [ MIME/Secondary = "mpeg" ];

4  %detector  ID3(Location);

5  %atom str  Title, Performer, Album, Genre;
6  %atom int  Year;

7  Audio      : MP3Type MP3;
8  MP3        : ID3?;
9  ID3        : Title Performer Album Year Genre;
```

B.8 The Video Feature Grammar

```
1  %module    Video;

2  %use       WWW

3  %detector  VideoType [ MIME/Primary = "video" ];

4  WebBody    : VideoType Video;
```

B.9 The MPEG Feature Grammar

```
1  %module    MPEG;

2  %use       Video;

3  %detector  MPEGType [ MIME/Secondary = "mpeg" ];
4  %detector  Icon(Location);

5  Video      : MPEGType MPEG;
6  MPEG       : Icon;
```

B.10 The Acoi Feature Grammar

```
1  %module    Acoi;

2  %use       WWW;
3  %use       Text, HTML;
4  %use       Image;
5  %use       Audio, MIDI, MP3;
6  %use       Video, MPEG;
```

B.11 The Tennis Feature Grammar

```
1  %module    Tennis;

2  %use       Video;

3  %atom flt  xPos, yPos, Ecc, Orient;
4  %atom int  FrameNo, Area;

5  %detector  xml-rpc::Segment(WebObject/Location);
6  %detector  xml-rpc::Tennis(WebObject/Location,
7               ancestor::Scene/Begin/FrameNo,
8               ancestor::Scene/End/FrameNo);

9  %detector  Netplay  [ some $Player in Player satisfies
10                            $Player.yPos <= 170
11                     ];

12 %detector  Rally(parent::Tennis);

13 Video      : Segment;
14 Segment    : Scene*;
15 Scene      : Begin End Type;
```

```
16 Type        : "tennis" Tennis;
17 Type        : "closeup";
18 Type        : "audience";
19 Type        : "other";
20 Begin       : FrameNo;
21 End         : FrameNo;
22 Tennis      : Frame+ Event;
23 Frame       : FrameNo Player;
24 Player      : xPos yPos Area Ecc Orient;
25 Event       : Netplay? Rally?;
```

B.12 The Australian Open Feature Grammar

```
1 %module    AO;

2 %use       Acoi;
3 %use       Tennis;
```

B.13 The Rijksmuseum Feature Grammar

```
1  %module    Rijksmuseum;

2  %use       Image;

3  %atom flt  threshold;
4  %atom int  columns, rows;
5  %atom int  column, row;
6  %atom int  x, y, width, height;
7  %atom flt  dark_coverage, light_coverage;
8  %atom int  number;
9  %atom flt  corr, non_corr, norm_corr;
10 %atom flt  scalar;
11 %atom bit  onoff;

12 %detector  light(Location);
13 %detector  global(WebObject/Location);
14 %detector  contrast(WebObject/Location);
15 %detector  grid(WebObject/Location);
16 %detector  histo_segment(Location);
17 %detector  segment(Location,
18               ancestor::light/histo_segment/threshold);
19 %detector  light_dist(ancestor::image/general/light/segment/name,
20               shape);
21 %detector  region(grid);
22 %detector  co_histo(WebObject/Location);
```

```
23 %detector  cu_histo(WebObject/Location);
24 %detector  im_histo(WebObject/Location);

25 %classifier decrules::clair_obscur(corr,non_corr,contrast/scalar);
26 %classifier decrules::cubism(corr,non_corr,contrast/scalar);
27 %classifier decrules::impressionism(corr,non_corr,contrast/scalar);

28 Image       : general global local style;

29 general     : light;
30 light       : Location histo_segment segment;

31 histo_segment: threshold+;
32 segment     : Location;

33 shape       : bbox;
34 bbox        : x y width height;

35 features    : light_dist;
36 light_dist  : light_coverage dark_coverage;

37 global      : shape features contrast;
38 local       : grid region;

39 contrast    : scalar;

40 grid        : columns rows cell*;
41 cell        : column row shape features;

42 region      : number;

43 style       : co_histo clair_obscur;
44 style       : cu_histo cubism;
45 style       : im_histo impressionism;
46 style       : ;

47 co_histo    : corr non_corr norm_corr;
48 cu_histo    : corr non_corr norm_corr;
49 im_histo    : corr non_corr norm_corr;
```