



UvA-DARE (Digital Academic Repository)

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

Windhouwer, M.A.

[Link to publication](#)

Citation for published version (APA):

Windhouwer, M. A. (2003). Feature grammar systems. Incremental maintenance of indexes to digital media warehouses 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: <http://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Appendix B

Feature Grammars

B.1 The WWW Feature Grammar

```
1 %module      WWW;
2 %start      WebObject(Location);
3 %detector   WebHeader(Location);
4 %detector   Robot(Location, "AcoiRobot");
5 %detector   Explored(MIME);
6 %detector   Tried(parent::Status);
7 %atom       www::url {(^http://([^ :/]*)(:[0-9]*)?/?(.*)$
8               |(^file://(.*)$)};
9 %atom       temporal::date;
10 %atom      url Location;
11 %atom      date Modification;
12 %atom      lng Length;
13 WebObject  : Location (Robot WebHeader WebBody)? Status;
14 WebHeader  : Redirect? MIME Modification? Length?;
15 MIME       : Primary Secondary;
16 Status     : Explored | Tried;
```

B.2 The Text Feature Grammar

```
1 %module      Text;
2 %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

```

1  %module      Image;
2
3  %use        WWW;
4
5  %detector   ImageType [ MIME/Primary = "image" ];
6  %detector   Portrait [ Faces/Number = 1 ];
7
8  %detector   Global(Location);
9  %detector   Icon(Location);
10 %classifier decrules::Photo(Global);
11 %classifier decrules::Graphic(Global);
12 %detector   Skin(Location);
13 %detector   Faces(Location);
14 %detector   exec::Histogram(Location);
15
16 %atom vector::flts;
17
18 %atom int    Number;
19 %atom flt    Prevalent, Far, NormalizedFar, Saturation, Percentage;
20 %atom flts   HSB, RGB;
21 %atom bit    Animated;
22
23 WebBody      : ImageType Image;
24 Image        : Global Icon Class;
25 Global       : Size Color Animated;
26 Size         : Width Height Ratio;
27 Color        : Number Prevalent Neighbor Saturation Histogram;
28 Neighbor     : Far NormalizedFar;
29 Histogram    : RGB HSB;
30 Icon         : Location;
31 Class        : Graphic | Photo (Skin Faces Portrait?);
32 Skin         : Percentage;
33 Faces        : Number;

```

B.5 The Audio Feature Grammar

```

1  %module      Audio;
2
3  %use        WWW
4
5  %detector   AudioType [ MIME/Primary = "audio" ];
6
7  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;
```


