# Introducing XAIRA…

An XML-aware tool for corpus indexing and searching

**Lou Burnard**

**Tony Dodd**

*Research Technology Services, OUCS*

# Topics

- Background: from SARA to XAIRA
- Architectural issues
- What can you do that's *fantastic*?

http://www.oucs.ox.ac.uk/rts/xara/

# Software development: the conventional wisdom

i. Assess user needs/requirements
ii. Prototype systems to fit user needs
iii. Evaluate against user performance
iv. Repeat from stage ii. until either
    a) *user is happy, or*
    b) *money runs out*

# Software development: the usual practice

- Creeping featurism
  - hey, that's a cool idea, I'll bolt that on too
- The Hausmann effect
  - this is hopeless, we need to drive a few boulevards through here
- Modularity and standardized interfaces are your only friends

# Historical Background (c.1994)

- Original design goals
  - robust searching of very large (c. 1 Gb) amount of SGML data
  - re-use available indexing tools
  - usable by researchers in CL, NLP, lexicography
- Original assumptions
  - client/server architecture
  - index build once only
  - one specific corpus (the BNC) only

# Historical Background (c.2002)

- Design goals
  - robust searching of any amount of XML data
  - offload processing to other components wherever possible
  - assume nothing about input DTD
- Architecture
  - client/server still valid
  - expect to re-index often
  - expect multiple interfaces

# Why another search engine?

- Can't you do all this with Google?
  - Digital texts are not just for discovery and display
  - The methods of corpus linguistics have a wider relevance
- Can't you do all this with eXist?
  - Probably, but only if you have a team of programmers at your disposal!

# Xaira: the key features

- Supports word search, concordance generation and manipulation, collocation, lexical analysis
- Uses XML annotation to the max
- Supports XML-aware complex queries
- Leverages existing standards
  - TEI/XCES
  - Unicode
  - CSS and XML
  - SOAP (xmlrpc)
- Uses efficient and compact indexing appropriate to small or huge corpora

# Architectural issues

How do the various parts of a XAIRA system interact?

# First catch your corpus…

- any collection of well-formed XML documents
  - if a DTD  is supplied, the corpus must be valid
  - if no TEI header is present, one will be created
- the more you put in, the more you get out
- "texts" are defined independently of file structure, as are the relevant units within them
- all indexing information is stored in the corpus header

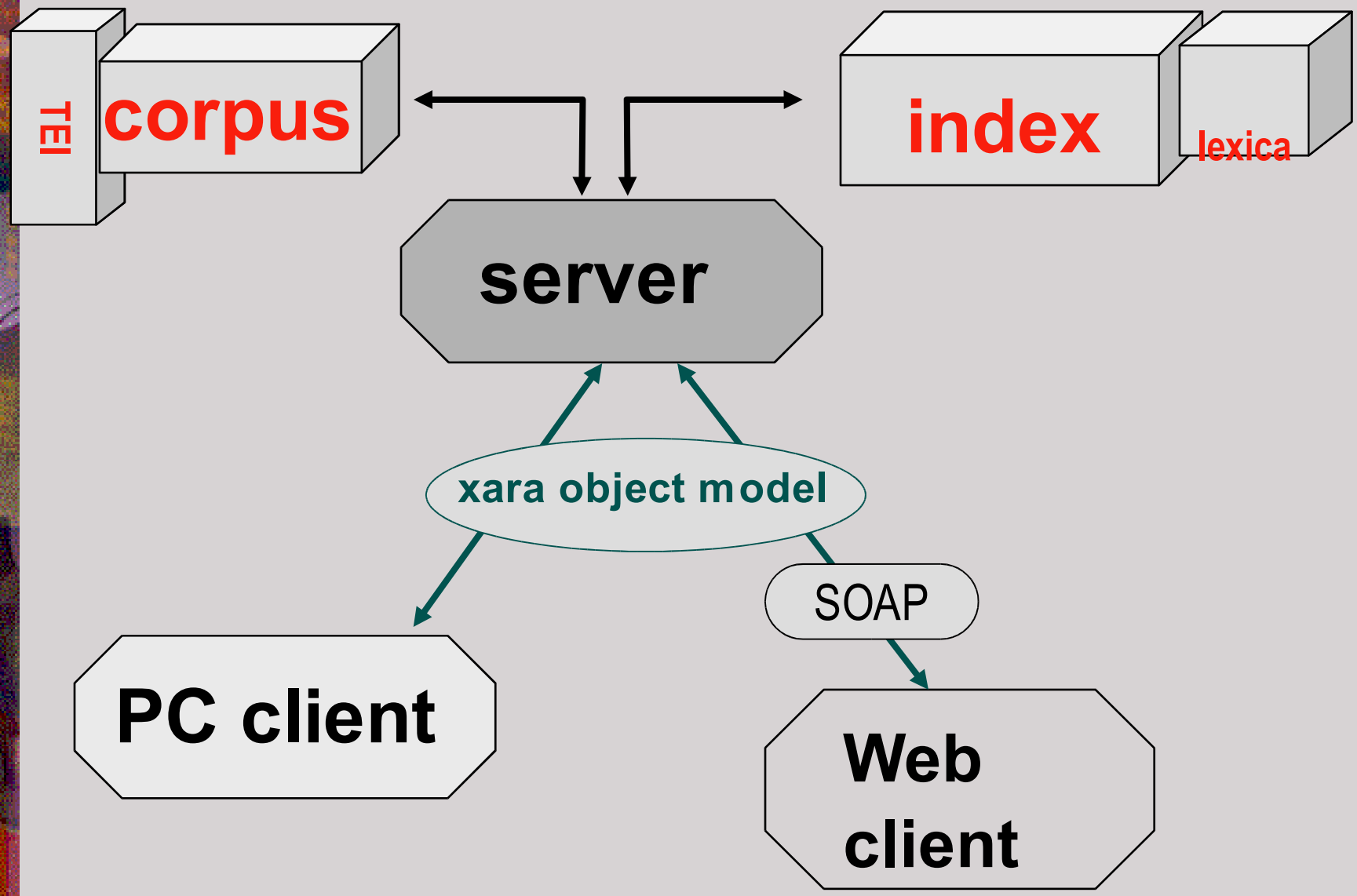
# Building the indexes

- tokenization
  - implicit, following Unicode rules (locale-sensitive)
  - explicit, following mark up
  - supports lexical features (eg collocation)
- lemmatization and POS tags
  - special case of "additional key" mechanism
  - generalized to provide fast context-specific searches
- tag indexes
  - attribute values and codebooks

# Next, build your index…

- Can be done simply by adding appropriate declarations to the TEI Header and running the indexer utility
- But probably easier to do with the supplied *Indextools* utility which
  - organizes and validates the files you are using
  - updates (or creates) the header with
    - *tokenization and indexing rules*
    - *tag and attribute usage, descriptive codebooks etc.*
    - *"bibliographic" metadata*
    - *default behaviour for character encoding, formats used, etc*
  - optionally runs and tests the indexer

# Architecture



# Hoorah for Unicode

- All data is held internally as Unicode
  - this allows us to defer most problems (e.g. tokenization, case-folding, line-breaking, character normalization, glyph composition) to someone else!
- User interface issues
  - For output, use one or more appropriate fonts
  - For input, we provide a keyboard definition utility

# Client/protocols

- The original SARA protocol
  - Corpus Query Language
  - Ad-hoc ASCII strings
- Now revised completely
  - Sara Object Model can be accessed
    - *directly by the client*
    - *via a SOAP wrapper*
    - *using saraScript*
  - The model defines
    - *CQL in XML*
    - *methods to manipulate CQL queries and results*
  - Support for web services

# Corpus Query Language

- Tokens
  - word, punctuation mark, substring
  - word+annotation/s (e.g. POS)
  - Unicode-compliant regular expressions for words, attribute values
  - XML start- or end-tag, plus attributes
- Boolean operations
  - negation, optionality
  - sequence, disjunction, join
- Scoped operations
  - within span, within XML element

# Client features

- User-configurable display
  - plain, XML, user-defined stylesheets
- User-definable keyboard mapping
- Texts, Results, Browse windows
- Results can be exported in XML
- Scripting language

# What can you do that's *fantastic*?

A sketchy over view of Xaira's query and display facilities

# Target queries

- What is the most frequent noun in this corpus?
- Find a random sample of 100 instances of "fish" followed by "chips" within 4 words
- Find sentences beginning with a conjunction.
- Show all inflected forms of the name "Winston".
- Show sentences which begin with "well" and end with a question mark.
- How often and in what contexts is the word "nature" used in different kinds of writing?
- Which verbs collocate significantly with "bosom" at different periods of history?
- Do men use colour vocabulary differently from women?

# Phrase or simple query

- search word or phrase
- can be case sensitive
- can include punctuation
- can include *anyword* character
- watch out for tokenization problems

SARA [Quirys]

File Edit Query View Windows Help

swa swa

m mode Criste þenigan æt his halgum weofode, swaswa eowrum hade gerist. Forþanþe ge synd g
wunode on þyssere worulde butan æltere synne swaswa nan oþer man.
niað on þam marinum þe næfre wifes ne brucað, swaswa he wunað on þam wifmarinum þe næfre
and wunedon on clænnysse, Criste folgigende. Swaswa Petrus cwæð to Criste sylfum
On oþre we scolan don be ures drihtnes lare. Swaswa he sylf cwæð on his haligan godspelle
On þam lendenum is, swaswa we leornigað on bocum, seo fule galnys
and on eallum godnyssum, mannum to bysene, swaswa byrnende leohtfatu.
And Iohannes geseah, swaswa we sædon ær, þone hælend ymbgyrdne
man and modes þæt halige husel him geoffrian, swaswa he sylf getæhte ær his þrowunge.
era cwellera ehtnyssa, þe þa martyras ofslogan, swaswa we sædon ær.
num com se haliga gast of heofenum to eorþan swaswa byrnende fyr mid bradum liggette ofer C
fyr mid bradum liggette ofer Cristes apostolos, swaswa Crist him ær behet
hyra abbode; calle hyra þinge him doð gemæne, swaswa him dihr se abbod.
manega Godes þeowan on þam scoton hadum, swaswa us secgð se canon.
am þam mycclan sinoþe, forðanþe se hælend is, swaswa ge gehyrdan oft, soð man and soð God,
to singanne dæghwamlice urum drihtne to lofe. Swaswa se witega Dauid on his witegunge cwæð
nne, ægðer ge libbendum ge þam forðfarenum; swaswa we leornigað on bocum.
Man sceal mæssian mid mycelre clænnysse. Swaswa þa haligan dydan, þe we hatað confessor
And mid clænnysse Criste þenedon. Swaswa þa canones us cyþað openlice Quod mul
ingð, þæt eow man syn ne sy, þæt ge swa libban swaswa læwede men.
ac we myngiað eow, þæt ge clænnysse healdan, swaswa Cristes þegenas on godum geþingðum,
nas on godum geþingðum, Gode to cwemnysse, swaswa þa haligan dydon, þe we her beforan ræc
labo eis quod non peribit ; God soðlice gecwæð, swaswa us sæde se witega:
Ac Crist cwæð swaþeah swaswa her cwæð on leden, hæt synd sume men

Fct Help, press F1

# Word Query

- searches the lexicon for word stem or pattern
- returns matching word forms with frequencies
- can restrict by frequency
- can apply lemmatization rules
- then carries out a lookup to display hits

Word Query

so[þð].* U Lookup

Pattern

| Word | Frequency | Forms |
|---|---|---|
| soþ | 89 | 26 |
| soþfæst | 7 | 5 |
| soþfæstnes | 8 | 6 |
| soþlic | 1 | 1 |
| soþlice | 66 | 2 |
| soþra | 1 | 1 |
| soþre | 1 | 1 |
| soþsagol | 1 | 1 |
| soþsagu | 1 | 1 |

| Form | pos | Frequency |
|---|---|---|
| soð | JJN | 19 |
| soðan | JJD | 10 |
| soþan | JJD | 7 |
| soðan | JJA | 6 |
| soðe | JJN | 6 |
| soða | JJN | 5 |

9 Words

Save Query

Show
Controls
Forms

Download Lemmata

A lemmatisation scheme determines how individual words are grouped under headwords. Select from the options below

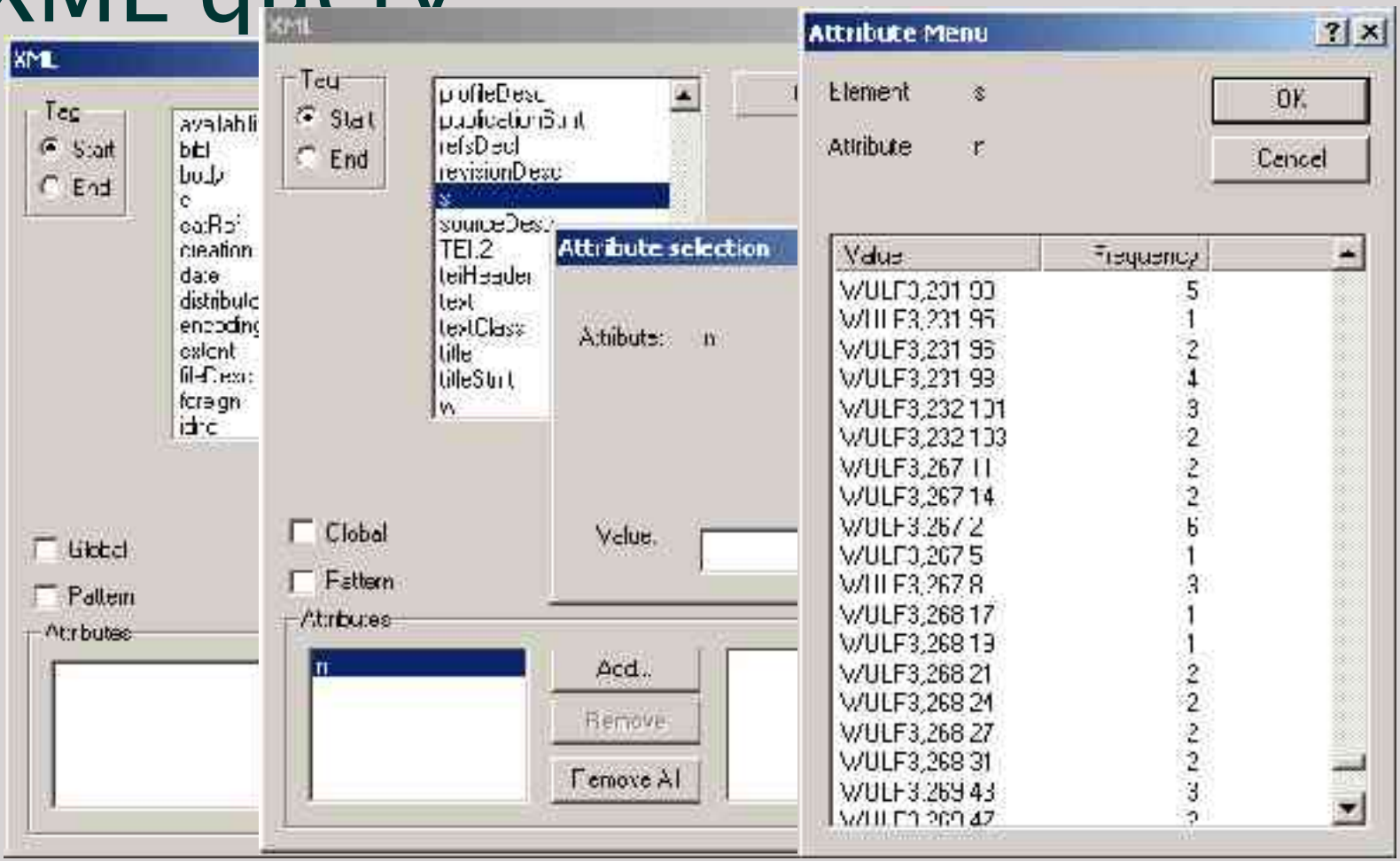
Apply

Headword

# XML query

- searches for XML start- or end-tags (not elements)
- start-tags optionally qualified by attribute values
- uses predefined codebooks (value indexes) if available
- useful in combination with other queries

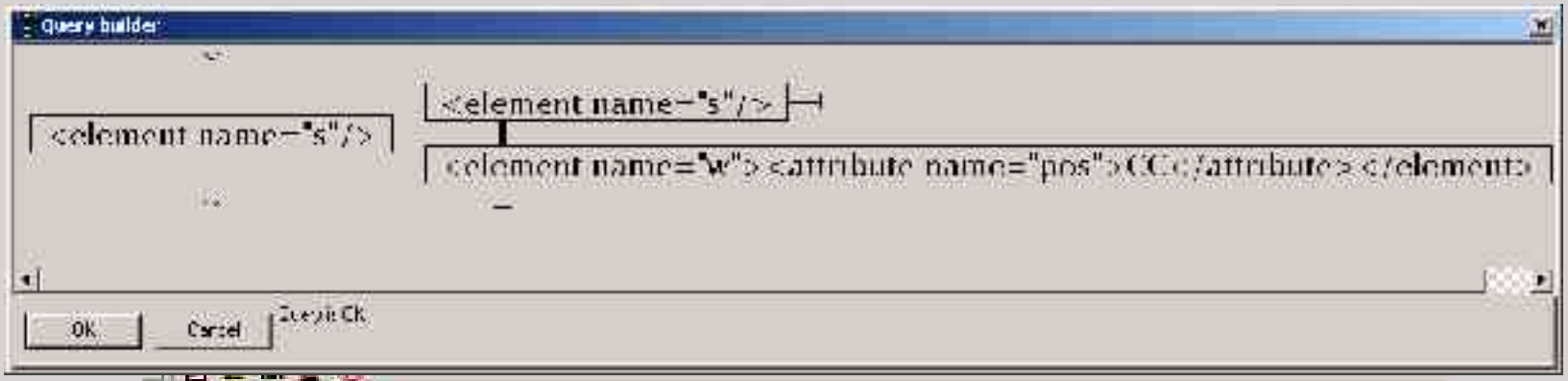
# XML query

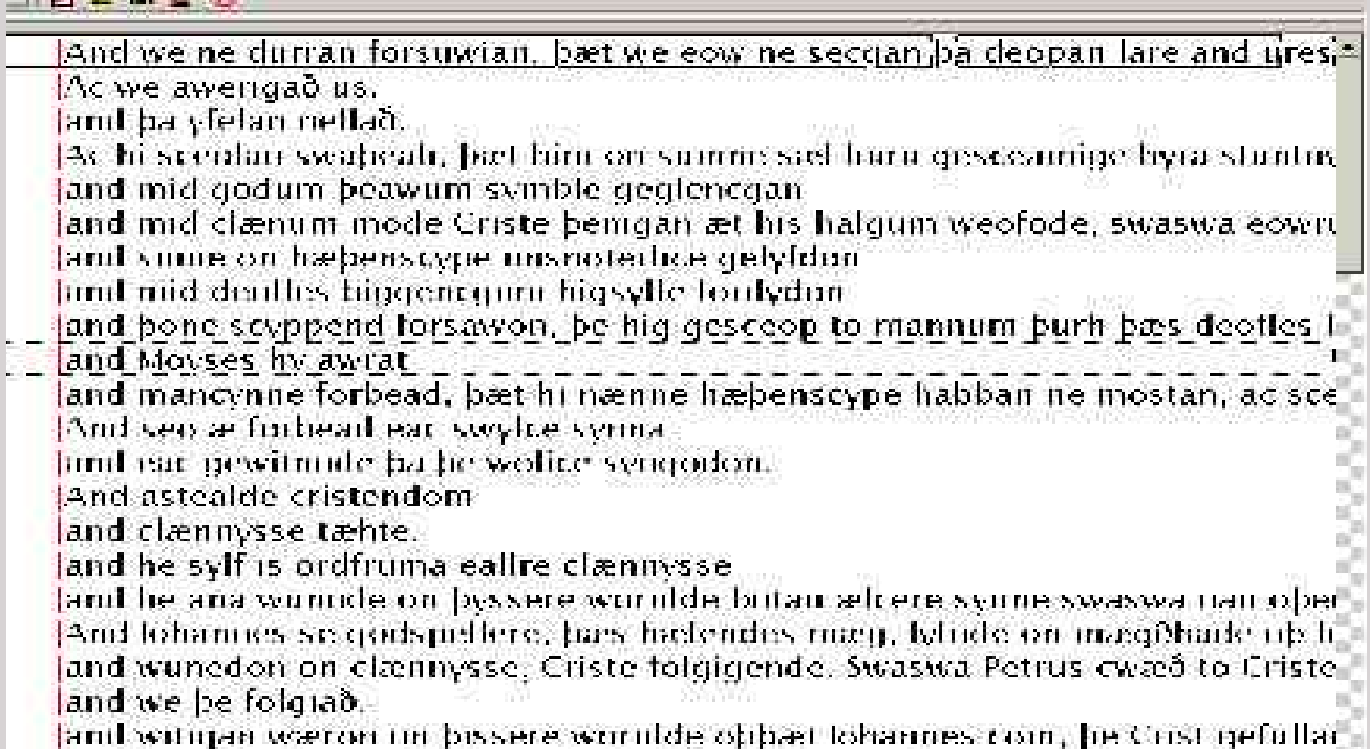

XML
Tag
Start
End
Global
Pattern
Attributes
Attribute selection
Attributes:
Value:
Add...
Remove
Remove All
Attribute Menu
Element
Attribute
OK
Cancel
Value
Frequency

# Building complex queries

- visual interface
- *scope node* defines where to look
  - an XML element
  - by span
- *query nodes* define what to look for
  - word, phrase, POS, pattern, XML, or AnyWord
- *link types* define sequence in which query node targets should occur
  - next, one-way, two-way

# Sentences beginning with conjunctions

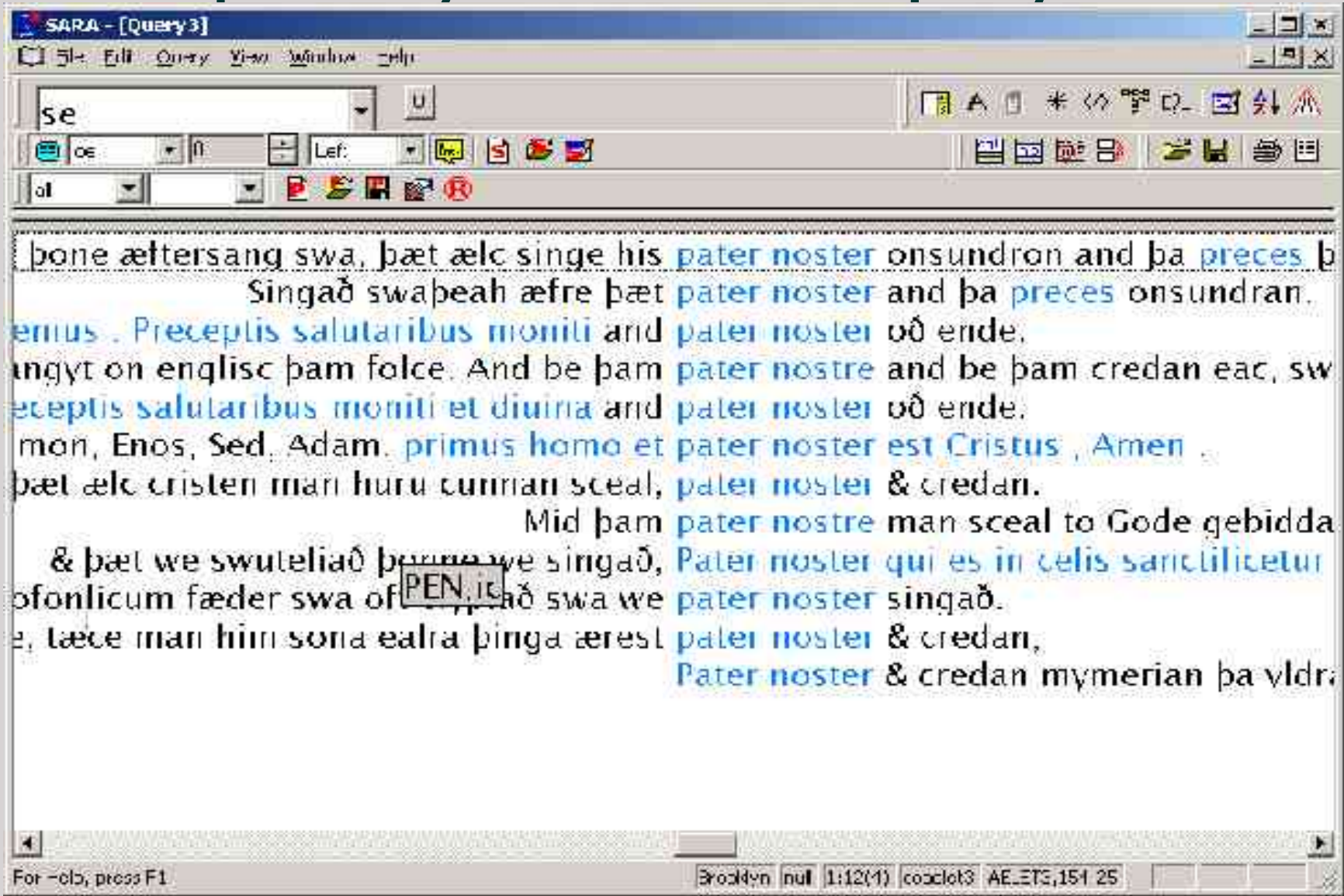


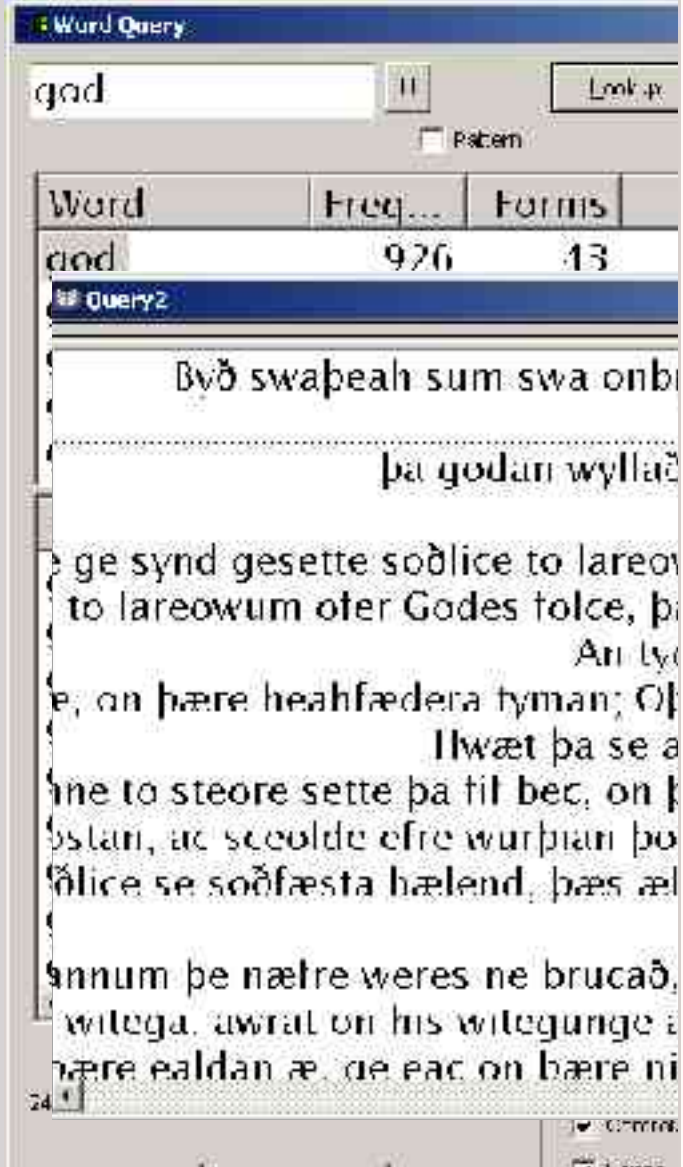
# Display of results

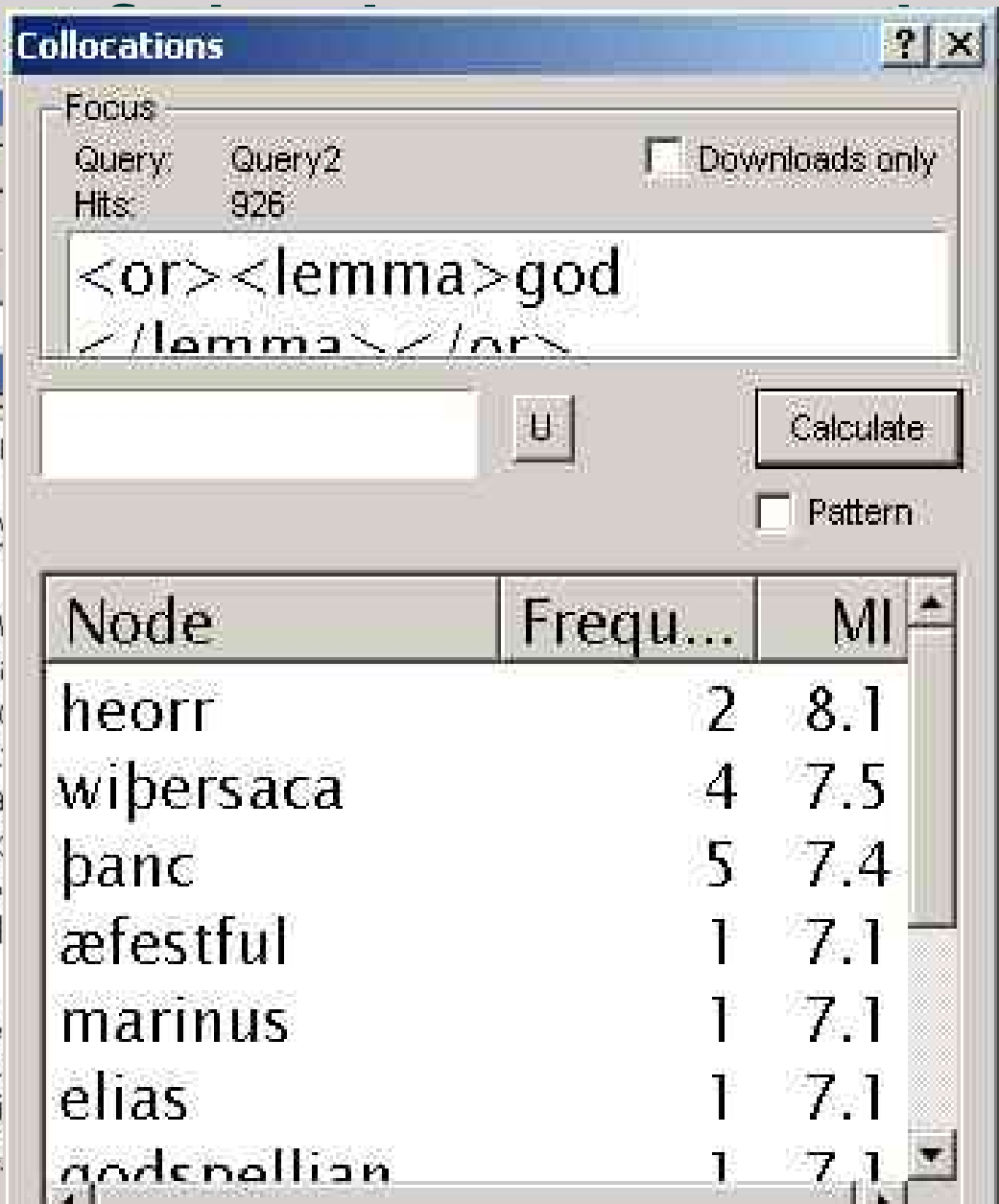
- Line (KWIC) or Page mode
- Context size expandable *ad lib*
- User defined formatting
  - stylesheet mechanism based on CSS
- Export of result files
  - in XML, or tab delimited

# Sample stylesheet display



# Collocations



# Manipulation of results

- Sorting
  - by left, right, or centre spans
  - by orthographic form or POS code
  - case sensitive or insensitive
- Thinning
  - at random
  - by selection
- Analysis and partitioning

# Partitions

- A partition is a way of grouping the texts making up a corpus, according to
  - *some explicit annotation or characterization (e.g. an attribute value)*
  - *according to whether or not they match a query (a partition of two halves)*
  - *arbitrary manual classification*
- Each member of a partition is a discrete text
- Analysis shows the rate of occurrence of hits within members of the partition
- Partitions can be saved and re-used or defined dynamically
- indextools generates a default partition using <catRef> element

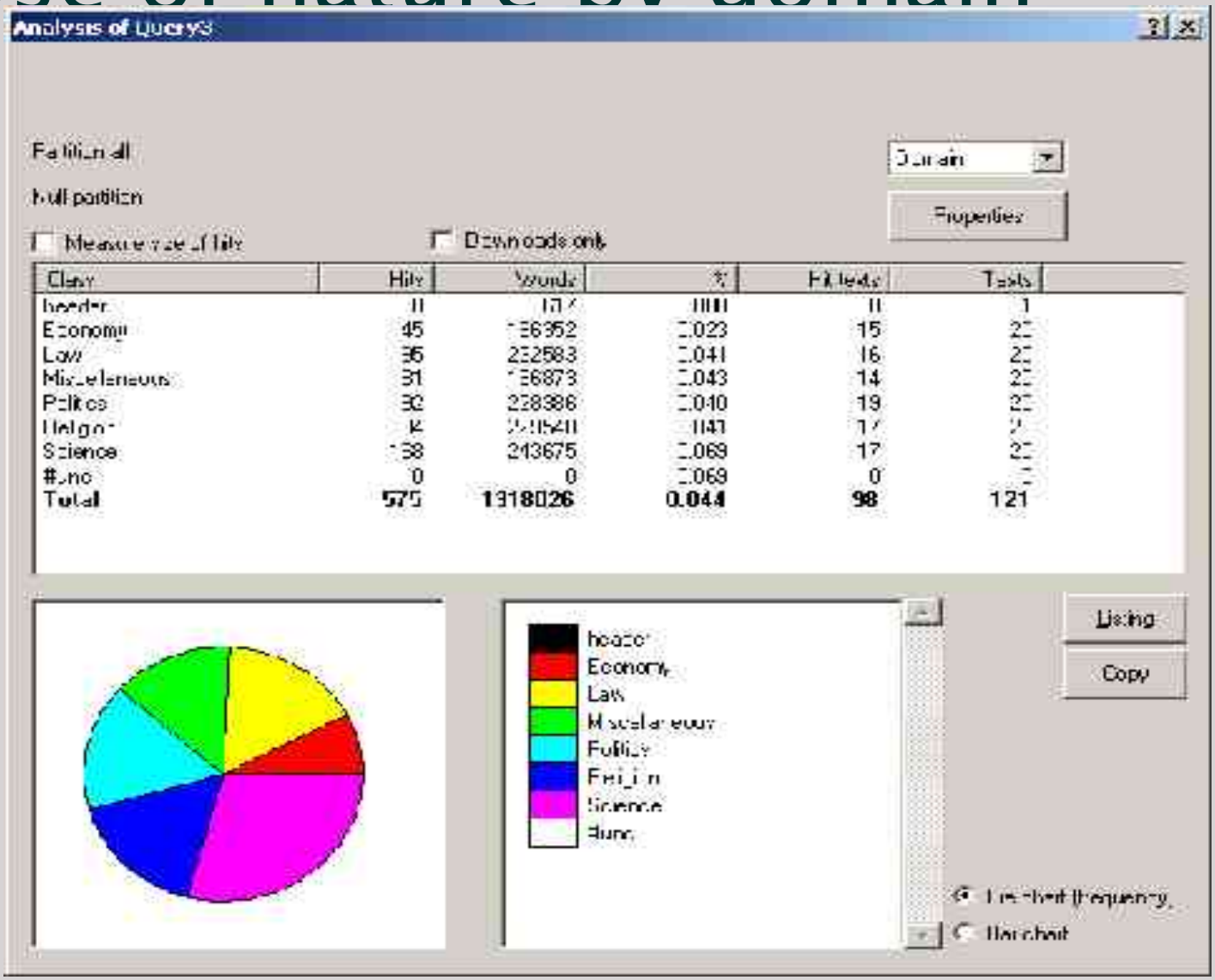
SARA - [Brooklyn]

File Edit Texts View Window Help

crist

| Text | Class | title | Query |
|---|---|---|---|
| coaelet3 | REL | Ælfric's First and Second Letter to Wulfstan | 25 |
| cowulf3 | HOM | Wulfstan's Homilies (O3) | 12 |
| coaelive | BLS | Ælfric's Lives of Saints | 8 |
| coaelet4 | REL | Ælfric's Letter to Wulfsige | 7 |
| cowulf4 | HOM | Wulfstan's Homilies (O3/4) | 6 |
| colaw2 | LAW | Alfred's Introduction to Laws, Alfred's Laws, Ine's Laws | 5 |
| cochroa2 | HIS | Anglo-Saxon Chronicles until 946 | 5 |
| colaw3 | LAW | Eleventh Century Laws | 5 |
| coorosiu | HIS | Alfred's Orosius | 1 |
| cogregd4 | BLS | Dialogues of Gregory the Great (MS C) | 1 |
| coboeth | PHI | Alfred's Boethius | 0 |
| cochroa3 | HIS | Anglo-Saxon Chronicles 951 - 1001 | 0 |
| coapollo | FIC | Apollonius of Tyre | 0 |
| cobede | HIS | Bede's Ecclesiastical History | 0 |
| header | header | Brooklyn Corpus | 0 |
| cogregd3 | BLS | Dialogues of Gregory the Great (MS H) | 0 |
| colaw4 | LAW | Late Laws | 0 |

# Use of nature by domain



# Saving and re-using queries

- Bookmarks
- Queries are saved with thinning information
- Optional annotation
- Associated bookmarks are preserved