xlinkit.com: consistency management as an application service

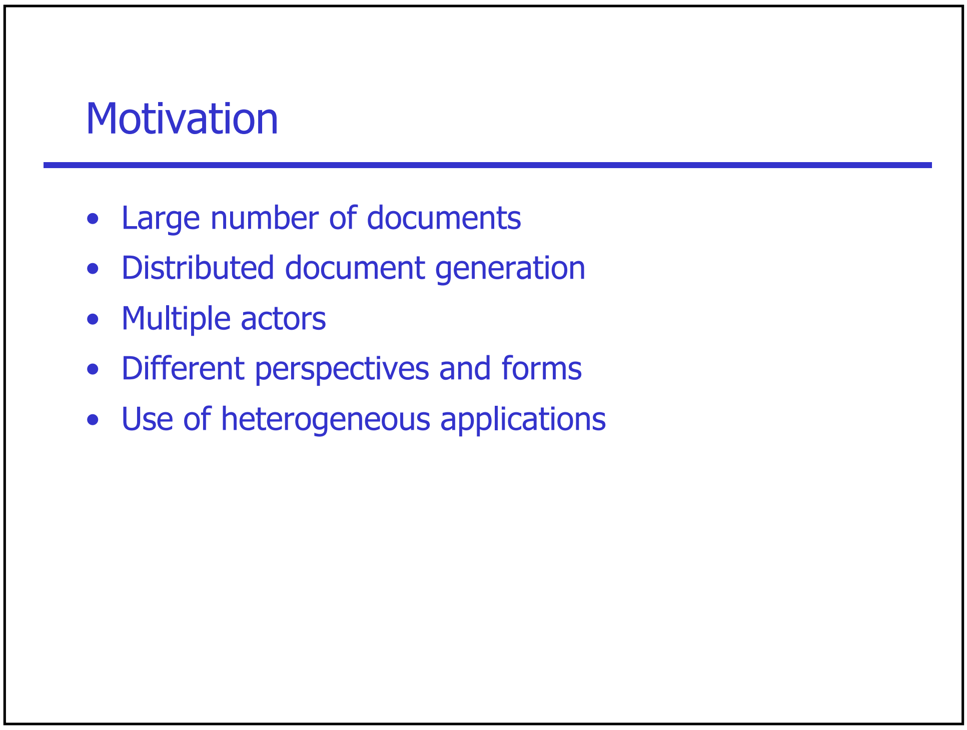
Anthony Finkelstein
University College London
Department of Computer Science

a.finkelstein@cs.ucl.ac.uk
http://www.cs.ucl.ac.uk/staff/A.Finkelstein
Also: http://www.consistencycheck.com

Outline

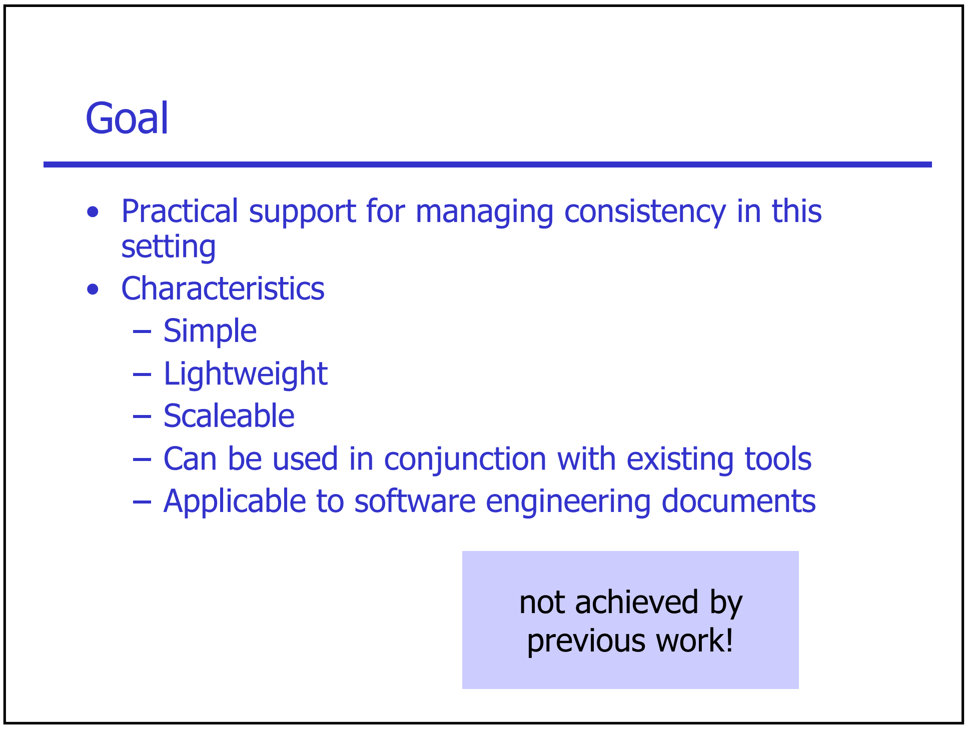
- Motivation
- A short technical tutorial
- Some working examples
- The rule language in a little more detail
- Architecture
- Performance and related issues
- Future work

Mostly a demo!



Motivation

- Large number of documents
- Distributed document generation
- Multiple actors
- Different perspectives and forms
- Use of heterogeneous applications



Goal

- Practical support for managing consistency in this setting
- Characteristics
  - Simple
  - Lightweight
  - Scaleable
  - Can be used in conjunction with existing tools
  - Applicable to software engineering documents

not achieved by
previous work!

Approach

- Check elements of distributed documents against "consistency rules"
- Link these elements in a manner dependent on whether the rules hold or not

realisation - a general purpose link generation engine

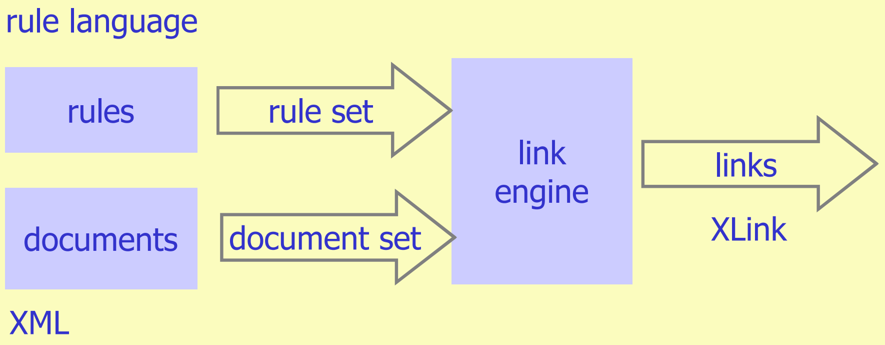
An accelerated tutorial

- XML
- XSL
- XLink
- XPointer
- A few other related helpings of alphabet soup
  - Cocoon
  - X2X
  - UML
  - XMI

xlinkit.com

- xlinkit.com is a lightweight application service which provides rule-based link generation and checks the consistency of distributed documents and web content
- You tell xlinkit.com the information you want to link and rules that relate the information. xlinkit.com will generate the links that you can then use for navigation. It will also diagnose inconsistent information and, if you want, provide you links directly to the inconsistent items of information
- xlinkit.com will eliminate the work required to directly author links and keep them up to date as well as simplifying the management of the consistency of distributed documents and web content

xlinkit.com



xlinkit.com

And now a demo …

More about the rule language

- Bold uppercase letters (S, D) refer to element sets (e.g. the set of all courses, etc).
- Lowercase letters are elements of sets with the same letter in uppercase (e.g. s is an element of S).
- The dot-notation refers to relative context, e.g. s.A is attribute A of element s (e.g. s.A could be the code of the current course).
- |S| denotes the cardinality of number of elements of a set S.
- S/(formula) denotes the restriction of set S according to a formula.

More about the rule language

- Our basic rules will then be of the form:

```
FORALL s in S, |D/(formula)| = size
```

- A formula is an operator expression, e.g. equality, or a formula connected by the boolean connectives "and" or "or". The formula may test properties on two things:
  - A relative attribute of the current source element of the forall expression
  - A relative attribute of the destination element being compared

More about the rule language

- For example, we can express our curriculum example as follows:
  - Let S be the set of Course elements in StudyPlan/Year elements
  - Let D be the set of code elements in syllabus/identity elements
  - Let s.A be the Code attribute of a Course element
  - Let d.A be the text contained in the code element of the syllabuses

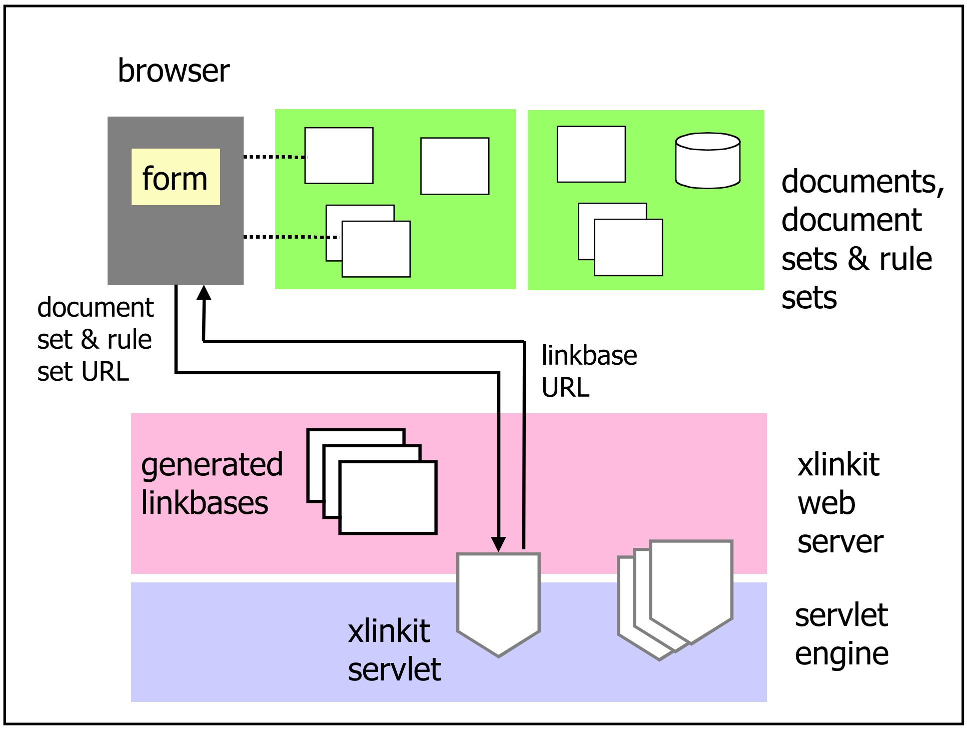
```
FORALL s in S, |D/(s.A=d.A)| > 0
```

Linking annotation

- generate consistent links
  - link the current element in the source set to every entry in the filtered destination set
- generate inconsistent links
  - every element in the source set will get an inconsistent link to the elements in the filtered destination set
- generate inconsistent links to a specified destination
  - generally link the current source element to the rule

Other features

- Rule language
  - Transitive closure operator
  - Pre-filtering
- Associated languages
  - DocumentSet markup
  - RuleSet markup



browser

form

documents, document sets & rule sets

document set & rule set URL

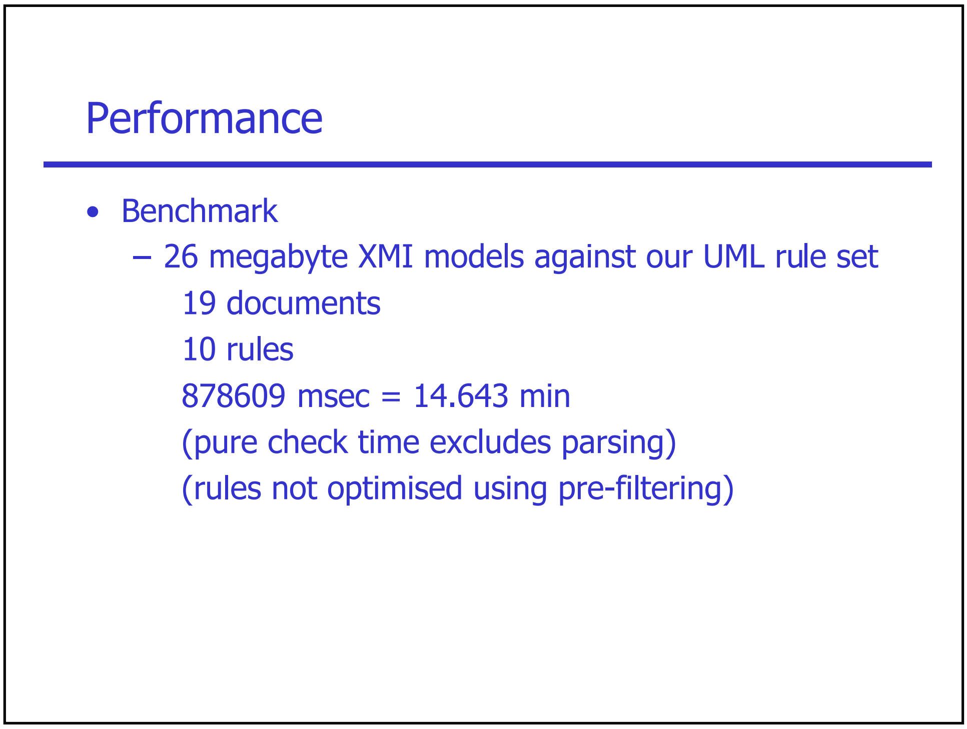
linkbase URL

generated linkbases

xlinkit web server

xlinkit servlet

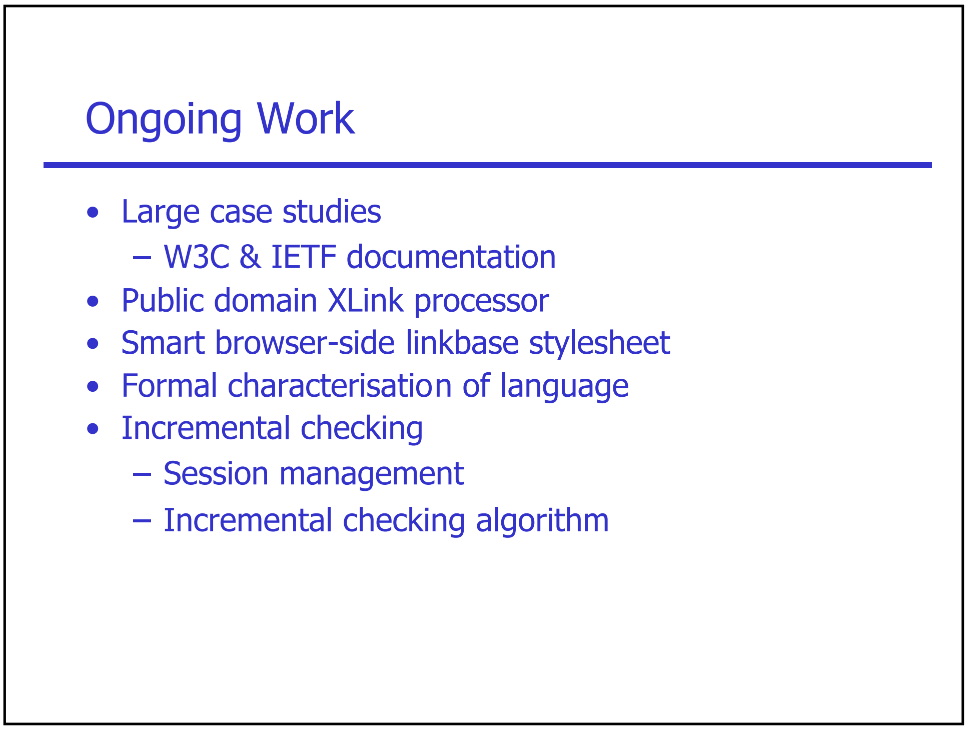
servlet engine



Performance

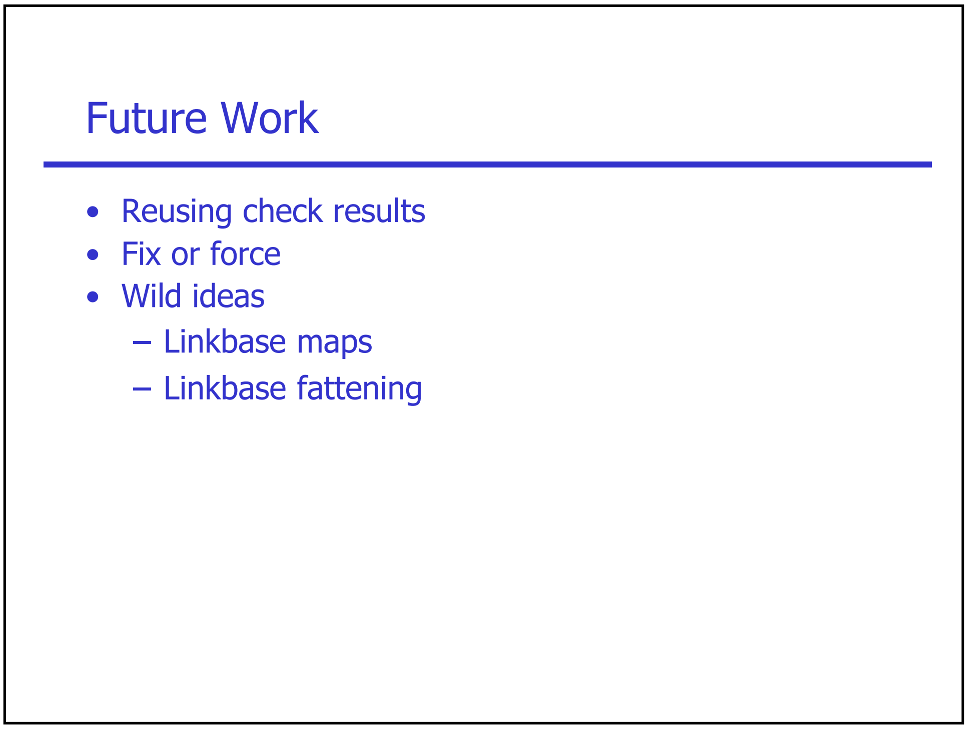
- Benchmark
  - 26 megabyte XMI models against our UML rule set

    19 documents
    10 rules
    878609 msec = 14.643 min
    (pure check time excludes parsing)
    (rules not optimised using pre-filtering)



Ongoing Work

- Large case studies
  - W3C & IETF documentation
- Public domain XLink processor
- Smart browser-side linkbase stylesheet
- Formal characterisation of language
- Incremental checking
  - Session management
  - Incremental checking algorithm



Future Work

- Reusing check results
- Fix or force
- Wild ideas
  - Linkbase maps
  - Linkbase fattening

With a little help from my friends

Licia Capra
Ernst Ellmer
Wolfgang Emmerich
Christian Nentwich
Torbjorn Revheim
Danila Smolko
Andrea Zisman

… and others!