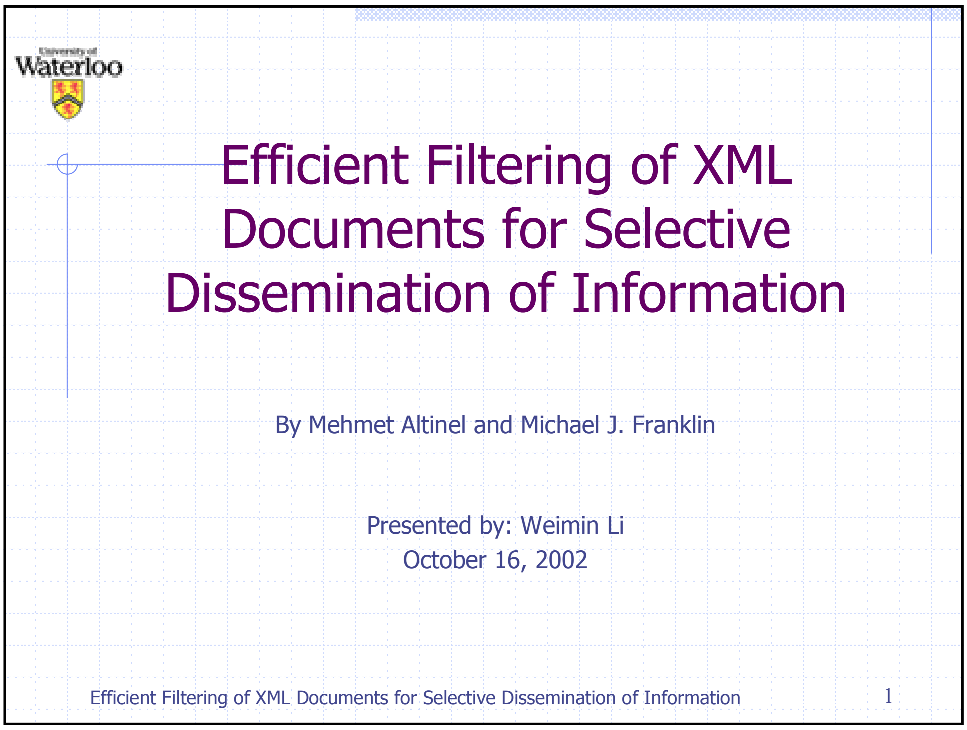
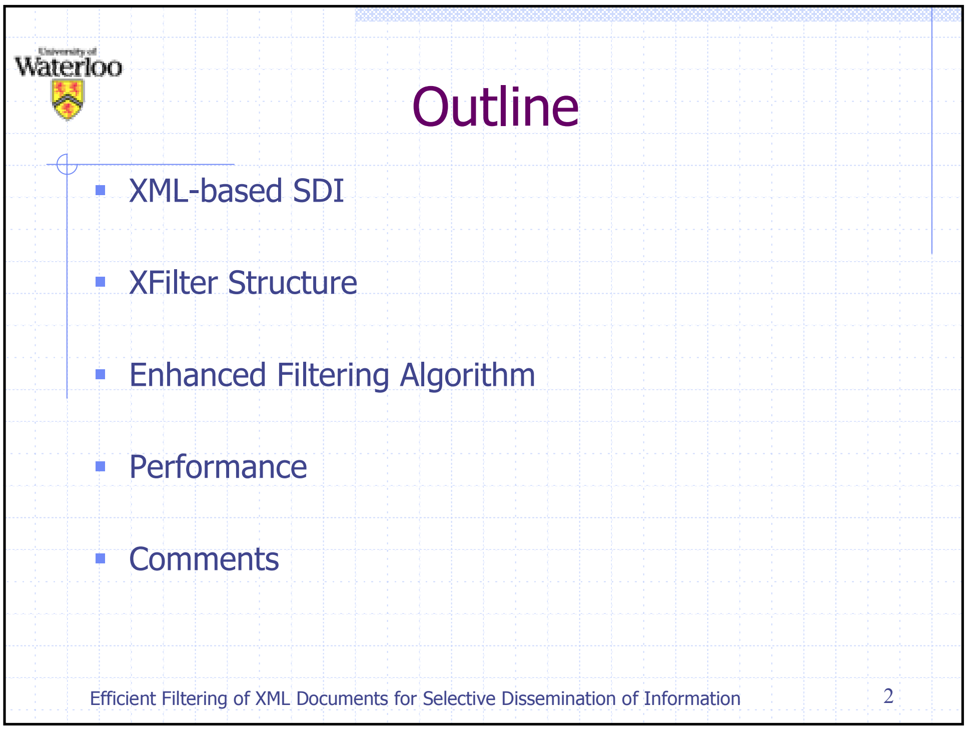


# Efficient Filtering of XML Documents for Selective Dissemination of Information

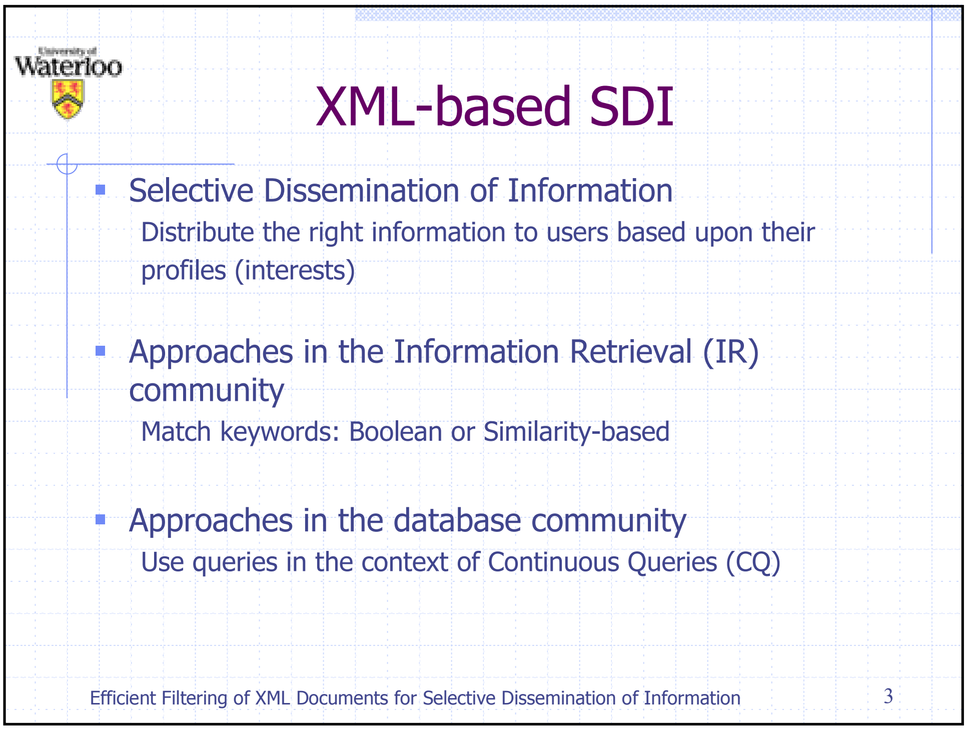
By Mehmet Altinel and Michael J. Franklin

Presented by: Weimin Li
October 16, 2002



## Outline

- XML-based SDI
- XFilter Structure
- Enhanced Filtering Algorithm
- Performance
- Comments



## XML-based SDI

- Selective Dissemination of Information

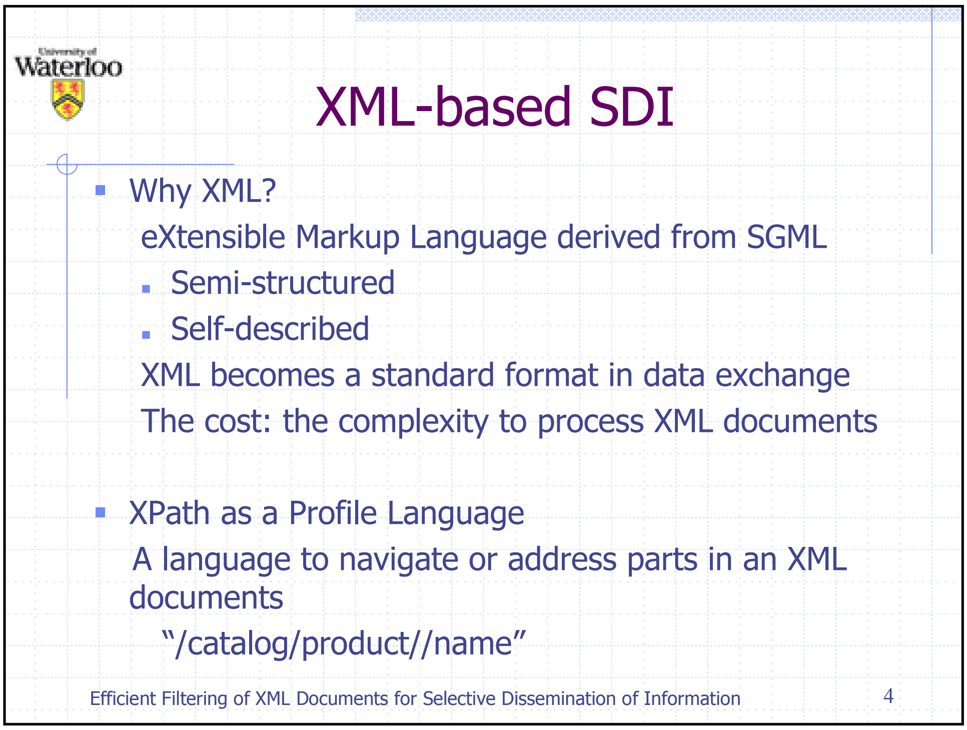
  Distribute the right information to users based upon their profiles (interests)

- Approaches in the Information Retrieval (IR) community

  Match keywords: Boolean or Similarity-based

- Approaches in the database community

  Use queries in the context of Continuous Queries (CQ)



## XML-based SDI

- Why XML?

  eXtensible Markup Language derived from SGML

  - Semi-structured
  - Self-described

  XML becomes a standard format in data exchange

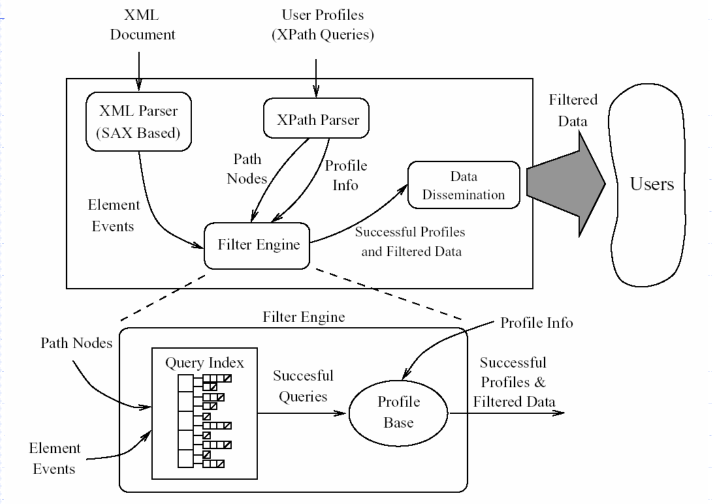
  The cost: the complexity to process XML documents

- XPath as a Profile Language

  A language to navigate or address parts in an XML documents

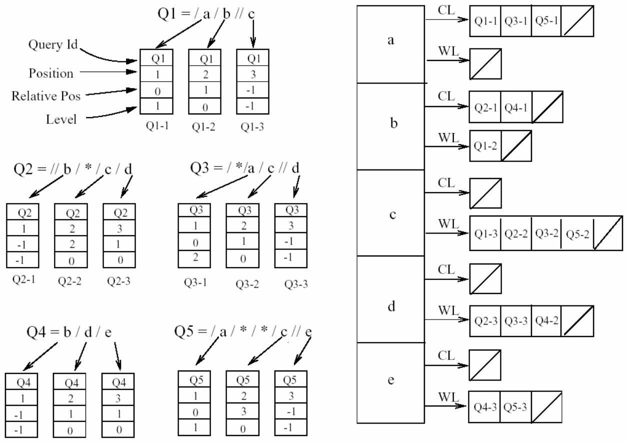
  “/catalog/product//name”

# XFilter Structure

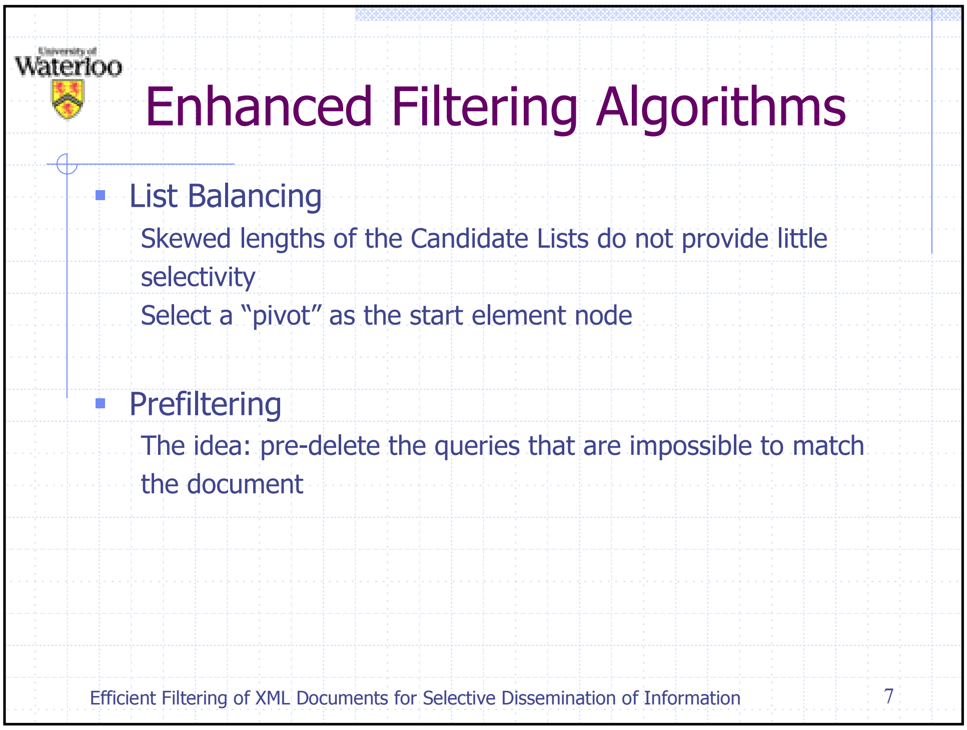


# XFilter

- Finite State Machine (FSM)



CL: Candidate List
WL: Wait List



# Enhanced Filtering Algorithms

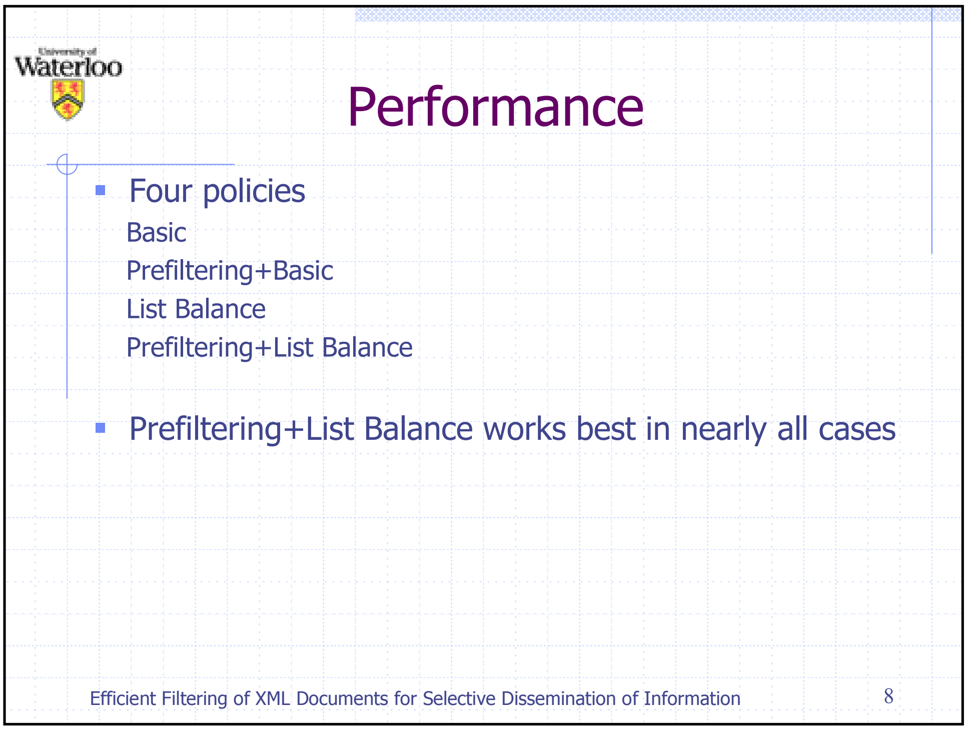
- List Balancing

  Skewed lengths of the Candidate Lists do not provide little selectivity

  Select a “pivot” as the start element node

- Prefiltering

  The idea: pre-delete the queries that are impossible to match the document



# Performance

- Four policies

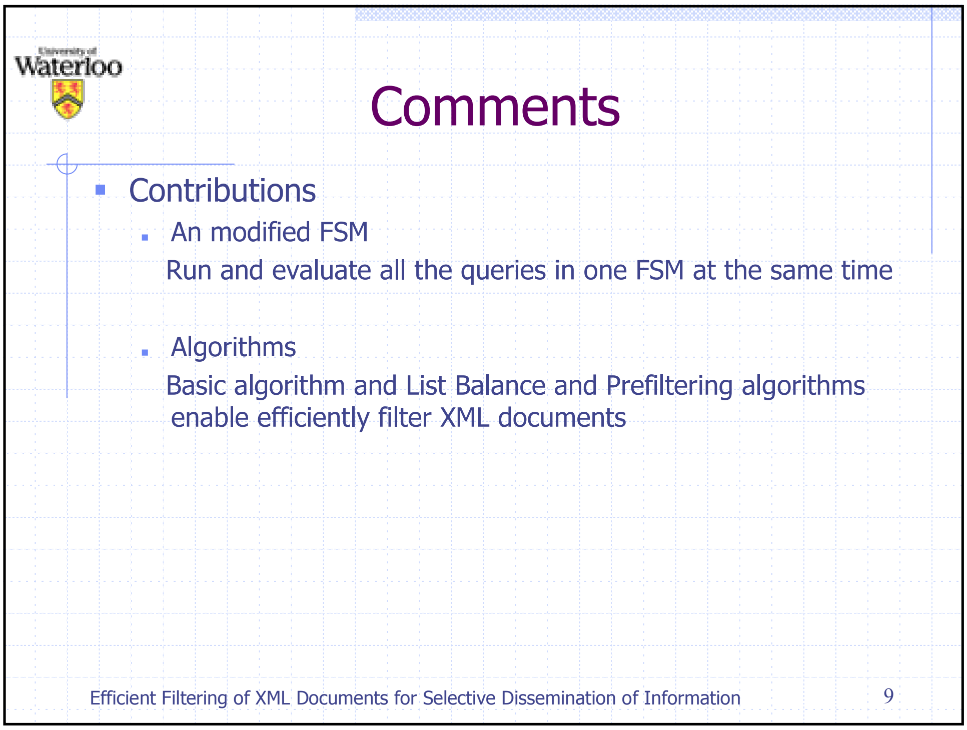
  Basic

  Prefiltering+Basic

  List Balance

  Prefiltering+List Balance

- Prefiltering+List Balance works best in nearly all cases



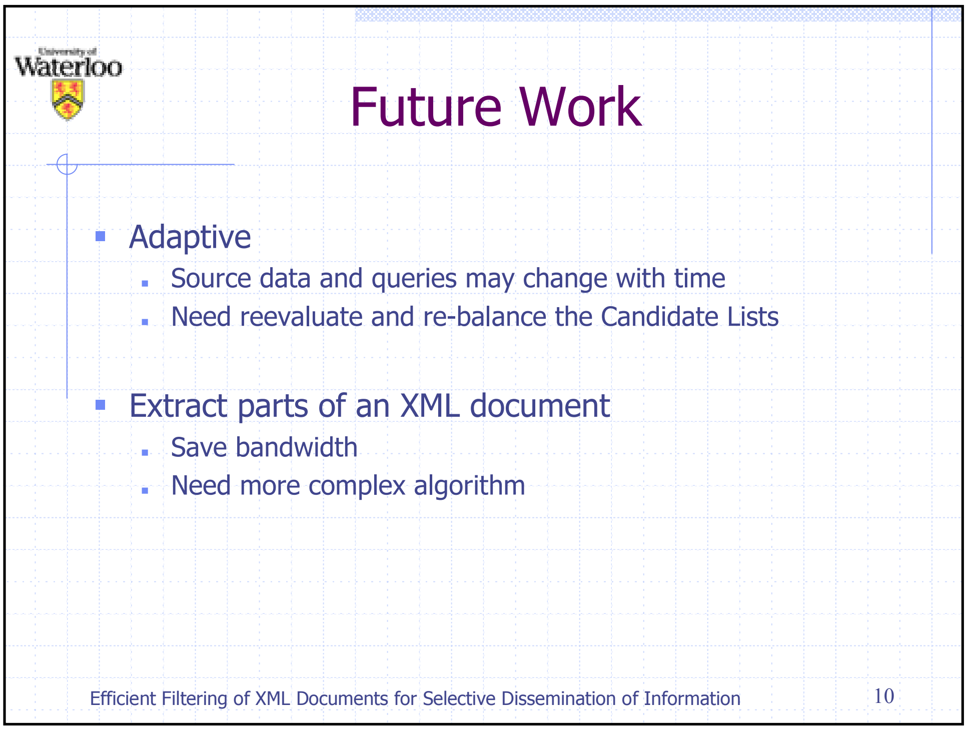
## Comments

- Contributions
  - An modified FSM

    Run and evaluate all the queries in one FSM at the same time

  - Algorithms

    Basic algorithm and List Balance and Prefiltering algorithms enable efficiently filter XML documents



## Future Work

- Adaptive
  - Source data and queries may change with time
  - Need reevaluate and re-balance the Candidate Lists
- Extract parts of an XML document
  - Save bandwidth
  - Need more complex algorithm