



# ExtMiner: Combining Multiple Ranking and Clustering Algorithms for Structured Document Retrieval

Miika Nurminen

Anne Honkaranta

Tommi Kärkkäinen

*Faculty of Information Technology*

*University of Jyväskylä, Finland*

# Motivation

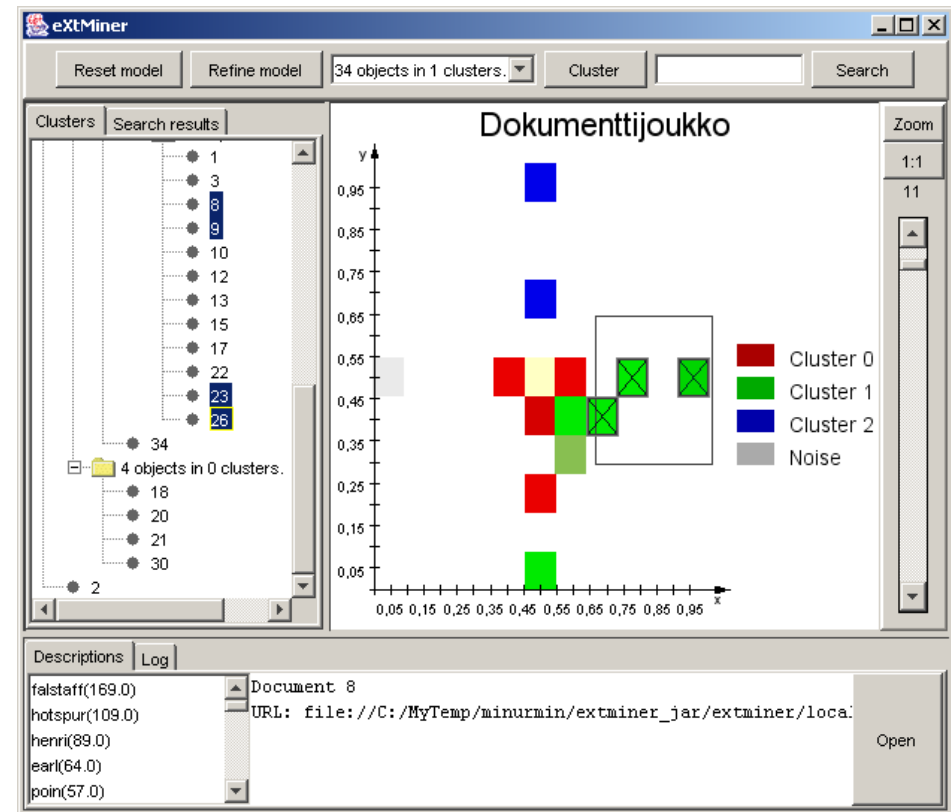
- Organizations are provided with overwhelming amount of digital information
  - New ways for retrieving, filtering and managing information are needed
- People find it difficult to express their information needs as index terms and keywords
  - Even if they do, the retrieved sets of documents do not necessarily match the information needs
- Heterogeneous document collections cannot be sufficiently searched when merely index terms are applied
  - (eg. Plain text vs. HTML vs. Word Doc vs. general XML)
- Potential solutions: integration of text mining techniques, providing different views to documents, taking document structure to account

# Related work

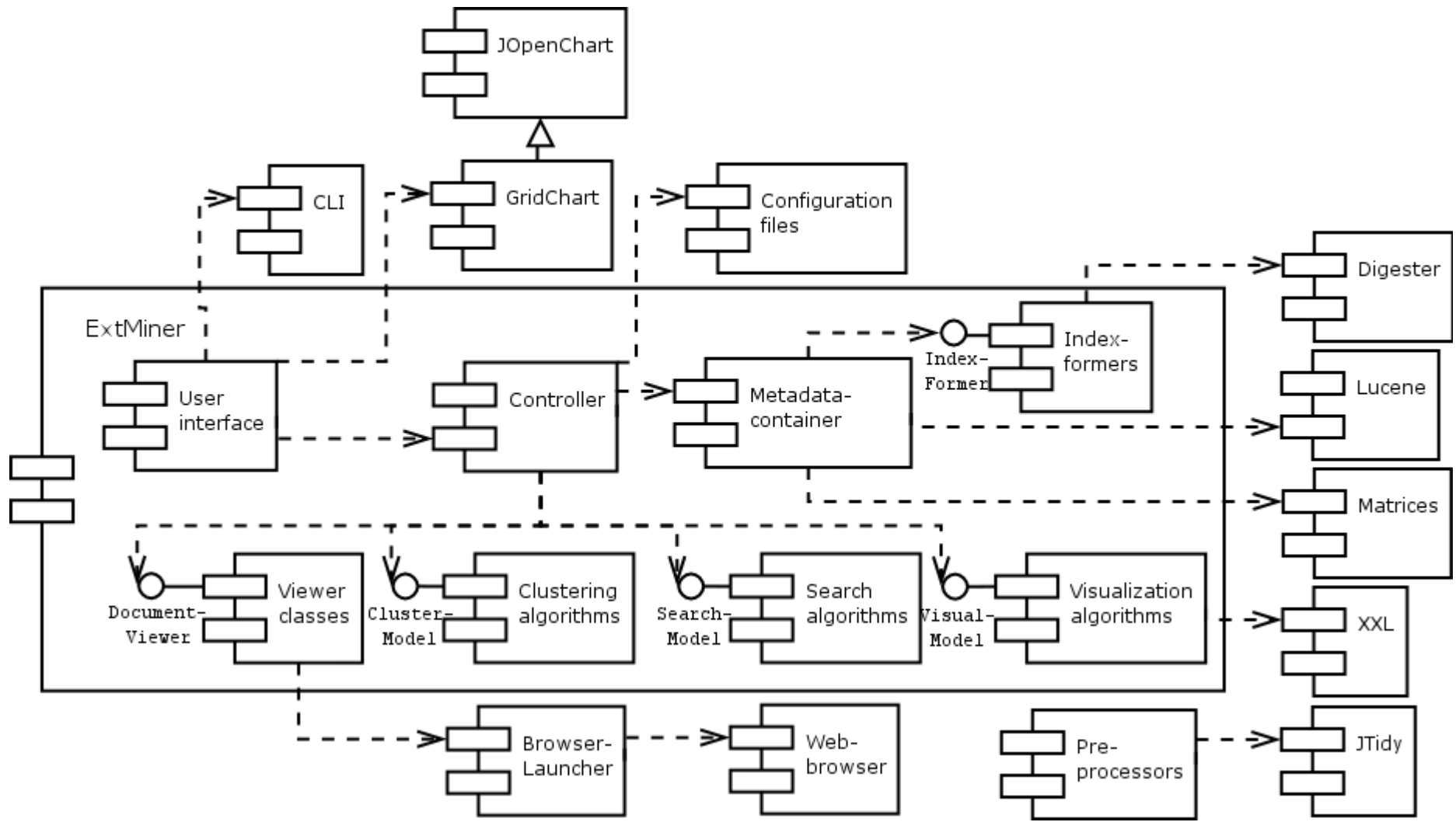
- **Extended Vector Model (Fox *et al*, 1988)** combines various document features (such as index terms and links) in ranking
- **Scatter/Gather-system (Cutting *et al*, 1992)** introduced continuous search process based on clustering
- **LightHouse (Leuski & Allan, 2000)** featured tight integration between ranked list and visualization of clusters
- **(Crouch *et al*, 2003)** have previously applied extended vector model for XML retrieval
- **MSEEC (Hannappel *et al*, 1999)** presented architecture for combining multiple clustering algorithms
- **(Ben-Aharon *et al*, 2003)** combined various rankers for content- and structure-based XML search

# Our approach: ExtMiner

- A platform and a proof-of-concept for combining
  - Different document features (eg. text, structure, links, metadata)
  - Ranking algorithms (eg. Cosine measure, PageRank)
  - Clustering algorithms (eg. DBSCAN, hierarchical clustering)
  - Visualization algorithms (eg. FastMap projection)
- Integrates many of the features previously implemented in separate systems
- Continuous search process based on ranked lists and cluster model



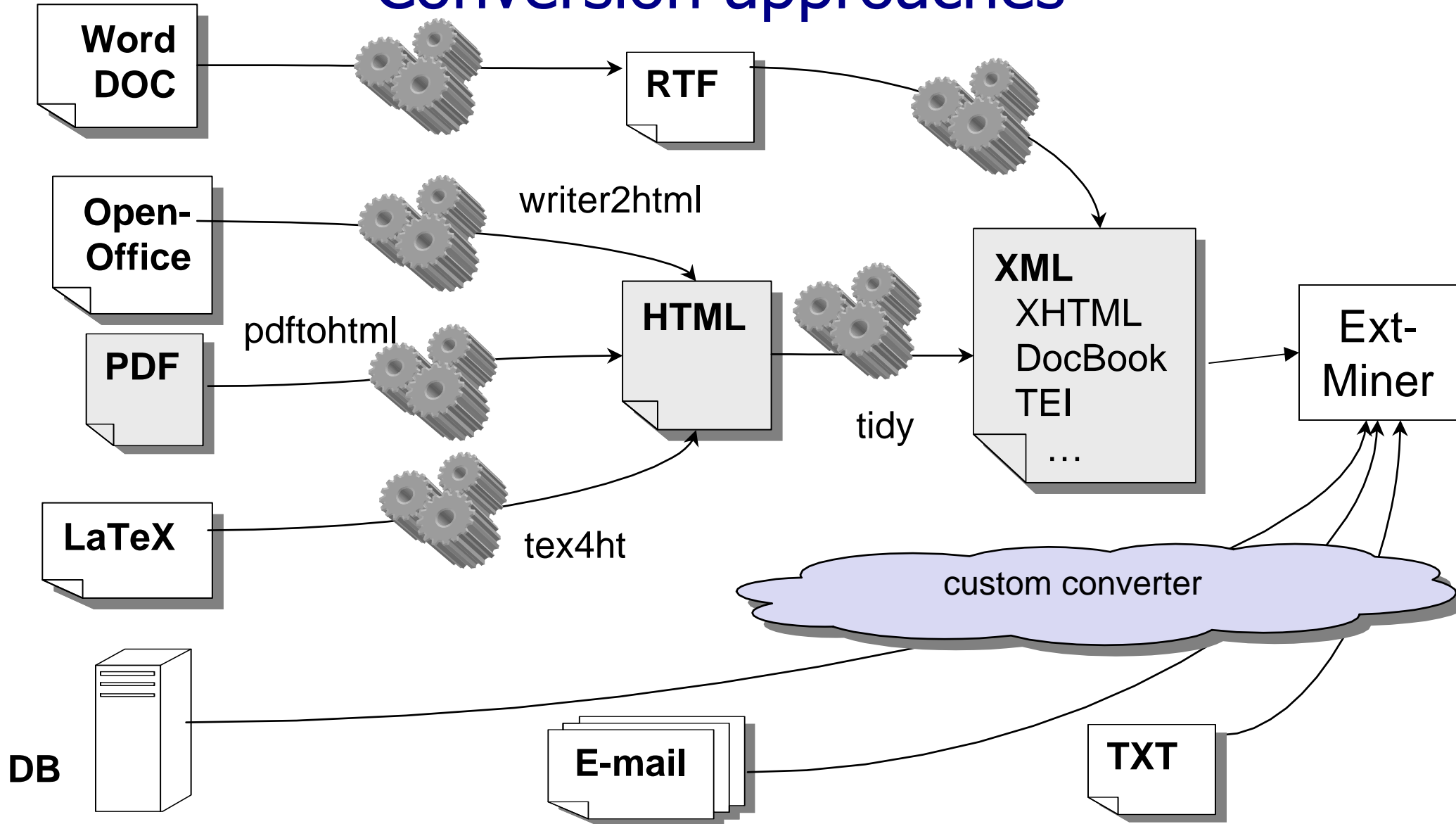
# ExtMiner architecture



# ExtMiner architecture (decomposed)

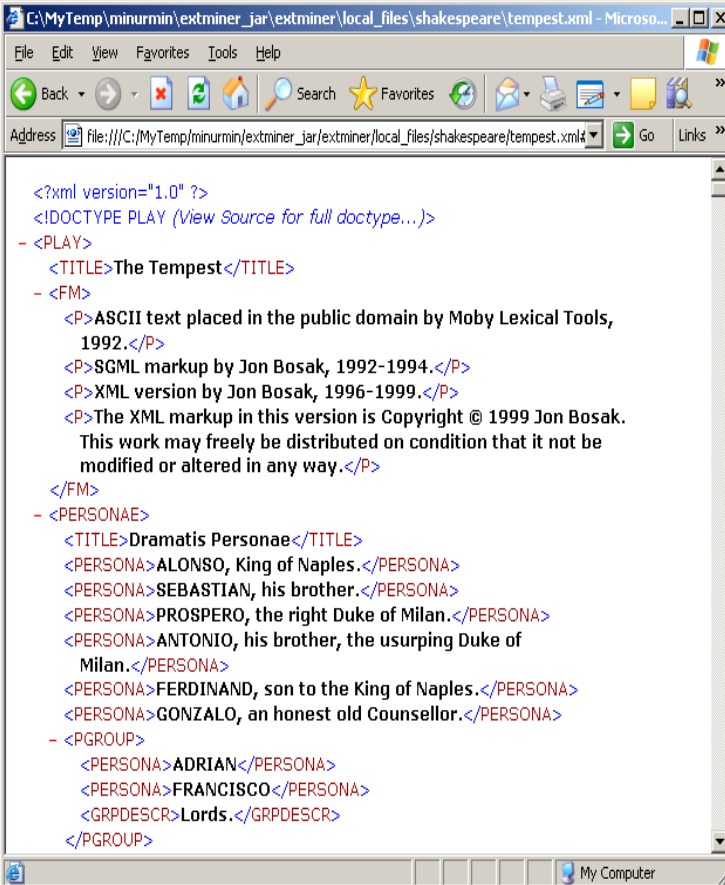
- 3 layers: UI, Application logic and Document index
- Document index consists of similarity matrices and a field-based term/link index
- Application logic includes pluggable ranking, clustering and visualization algorithms and extensible mechanism for index creation from various document repositories
- UI provides customizable views for documents, ranked search result list and cluster model tree
- Implemented with Java, published as open source. Third-party open source components (eg. Jakarta Lucene, JOpenChart) are utilized.

# Conversion approaches



# Indexing and configuration

- Documents must be available in a local filesystem
- Stemming, stopword removal and *tf\*idf* weighting is performed by Lucene
- Digester handles rule-based XML parsing
- Documents are represented as field-based index (eg. tuples of vectors)
- Fields can be index terms, links, headers or document type –specific external metadata or structural information encoded as vectors
- Document-to-document similarities are precalculated for clustering
- Different index formers and field definitions can be utilized, depending on document type and application domain



```
<?xml version="1.0" ?>
<!DOCTYPE PLAY (View Source for full doctype...)
- <PLAY>
  <TITLE>The Tempest</TITLE>
  - <FM>
    <P>ASCII text placed in the public domain by Moby Lexical Tools,
      1992.</P>
    <P>SGML markup by Jon Bosak, 1992-1994.</P>
    <P>XML version by Jon Bosak, 1996-1999.</P>
    <P>The XML markup in this version is Copyright © 1999 Jon Bosak.
      This work may freely be distributed on condition that it not be
      modified or altered in any way.</P>
  </FM>
  - <PERSONAE>
    <TITLE>Dramatis Personae</TITLE>
    <PERSONA>ALONSO, King of Naples.</PERSONA>
    <PERSONA>SEBASTIAN, his brother.</PERSONA>
    <PERSONA>PROSPERO, the right Duke of Milan.</PERSONA>
    <PERSONA>ANTONIO, his brother, the usurping Duke of
      Milan.</PERSONA>
    <PERSONA>FERDINAND, son to the King of Naples.</PERSONA>
    <PERSONA>GONZALO, an honest old Counsellor.</PERSONA>
  - <PGROUP>
    <PERSONA>ADRIAN</PERSONA>
    <PERSONA>FRANCISCO</PERSONA>
    <GRPDESCR>Lords.</GRPDESCR>
  </PGROUP>
```



# Searching and clustering

- Extended vector model is applied both in ranking and clustering similarity calculation
- Let  $d$  be a document and  $q$  a query, both represented as tuples of  $n$  vectors (fields). Relevance estimate  $R$  is calculated as

$$R(d, q) = \sum_{k=1}^n w_k \text{sim}_k(r_k(d), r_k(q))$$

- $r$  denotes the restriction that extracts  $k$ -th vector from the tuple,  $\text{sim}$  is the similarity measure (such as boolean matching, cosine measure or co-citation),  $w$  denotes a field-specific weight supplied by the user (or matched evenly by default)
- Substitute  $q$  with another document and you have a document-to-document similarity measure for clustering
- Any metric clustering algorithm can be used, provided that the implementation is available

# User interface and visualization

- Iterative search and clustering process
  - Search and clustering can be performed iteratively and focused to an appropriate subset of the collection
- Interactive cluster model
  - The user can select documents from any of the views provided by the application: ranked list, cluster tree or visual projection. Cluster tree is interactive: a cluster can be marked as noise or subclusters of a single cluster can be merged (useful with hierarchical clustering)
- Simultaneous views for lists and clusters
  - Both views are needed since lists and clusters support different search objectives. Clusters are easy to understand and help to cope with ambiguous terms, although they do not improve search quality as such.
- Any MDS (multidimensional scaling) –style projection algorithm can be used for visualization (currently FastMap)
- Documents can be opened in web browser or custom viewer (eg. text editor, XML tree view)

# Case 1: Course essays

- *Introduction to Software Engineering* –course was carried out in Fall 2004 at University of Jyväskylä
- Each student was assigned to produce 13 essays, one for each lecture. Over 200 signed up to the course, finally over 1000 essays.
- ExtMiner was utilized for checking up and comparing the essays
- Fields used: index term and headers were extracted directly from the documents. Author(s), major subject(s) and lecture number was provided as metadata.
- The lecturer could retrieve essays from the collection by using each of the fields as search key
- Clustering allowed cross-intersecting each cluster pertaining to certain lecture or subject matter in relation to each other

# Case 1: Course essays

eXtMiner

Reset model Refine model 78 objects in 3 clusters. Cluster +index:3 +majors:TII Search

Clusters Search results

- Pikki Vesa, 3 - 1.0
- Honkonen Tapio, 3 - 1.0
- Sinivuori Riikka, 3 - 1.0
- Kalmari Olli, 3 - 1.0
- Lehtinen Matti, 3 - 1.0**
- Koskenkorva Risto, 3 - 1.0
- Aho Kari, 3 - 1.0
- Huikari Juha, 3 - 1.0
- Alatalo Jani, K..., 3 - 0.82961982488632

Dokumenttijoukko

Zoom 1:1 23

Cluster 0 Cluster 1 Cluster 2 Noise

Descriptions Log

Document 42  
URL: <http://www.cc.jyu.fi/~makaleht/itk130/luento3.tx1>  
authors=Lehtinen Matti  
head=Ohjelmatuotannon keskeisiä ongelmia  
index=3  
majors=TIE

Open

Text viewer

Ohjelmatuotannon keskeisiä ongelmia  
Joonas Lampinen, Ville-Matti Pasanen ja Matti Lehtinen

Projektitoiminnan tarve:  
Projektitoiminta on vaikeaa, sillä projektissa on eri tasoilla toimivia tekijöitä, esim. suunnittelijat ja koodaajat. Tällöin voi tasojen välinen kommunikointi ja yhteistyö olla hankalaa. Lisäksi projektit voivat olla huomattavan suuria, jolloin projektin kehittäminen joudutaan pilkkomaan pienempiin ryhmiin. Useiden pienempien ryhmien saumaton yhteistyö voi olla hankalaa. Projektissa vaikuttavana tekijänä on myös työntekijöitten eritasoisuus.

Muutostarpeet:  
Vaikka tekniikka on kehittynyt nopeasti, ei ohjelmistotuotanto ole pysynyt mukana. Ohjelmistotekniikka on nuori ala ja vaikka keksittäisiinkin uusia menetelmiä, niitten testaaminen ja kehittäminen vie aikaa. Tällöin uudet menetelmät saattavat tulla käyttöön vasta 10-20 vuoden viiveellä.

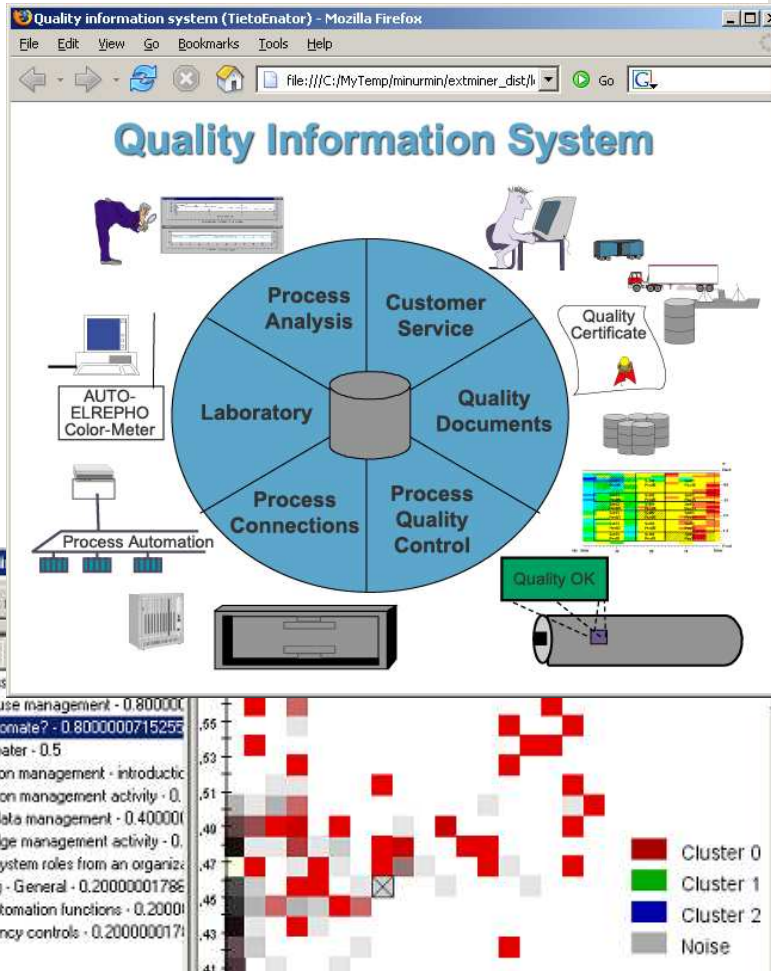
Integraation tarve:  
Erilaisten ohjelmistojen tuottaminen on erilaista, on vaikea löytää yleistettäviä tehokkuutta kasvattavia menetelmiä. Ohjelmistot ovat erilaisia ja niiden laatuvaatimukset voivat poiketa huomattavasti. Toisille ohjelmistoille on valmiita komponentteja, kun taas jotkin ohjelmistot vaativat aivan oman suunnittelun tehokkuuden takaamiseksi.

Tuotteen hallinta:

## Case 2: KnowPap

- KnowPap is an e-learning application for paper production technologies, containing a collection of HTML-documents, pictures, video clips and other education material
- A subset of 300 documents was imported from KnowPap web site and indexed in ExtMiner
- Index terms, headers and links to multimedia material (including target type) were extracted from HTML files
- With multimedia link index ExtMiner was used as a proof-of-concept interface for browsing a simple multimedia "database". The user could retrieve web pages or directly multimedia material, depending on query.
- Paper technology trainers could use ExtMiner as a tool for organizing, browsing and retrieving training material components for novel training content

# Case 2: KnowPap



**KnowPap - Why automate? - Mozilla Firefox**

File Edit View Go Bookmarks Tools Help


file:///C:/mytemp/minurmin/extminer\_dst/local\_files/knowpap/english/automation/1\_general/1\_why\_automation/frame.t

## Why automate?

### Improvement of product quality

Improving quality by means of automation requires the constant measurement and control of several quality parameters. Quality measurements usually require the use of specialized gauges and analyzers.

Automation is used to reduce the differences between shifts in continuous processes and otherwise run processes more evenly. When the consistency of product quality is improved, waste production and the percentage of overquality is reduced.



### Increase in production

Increase in production capacity achieved by automation may be the result of several factors.

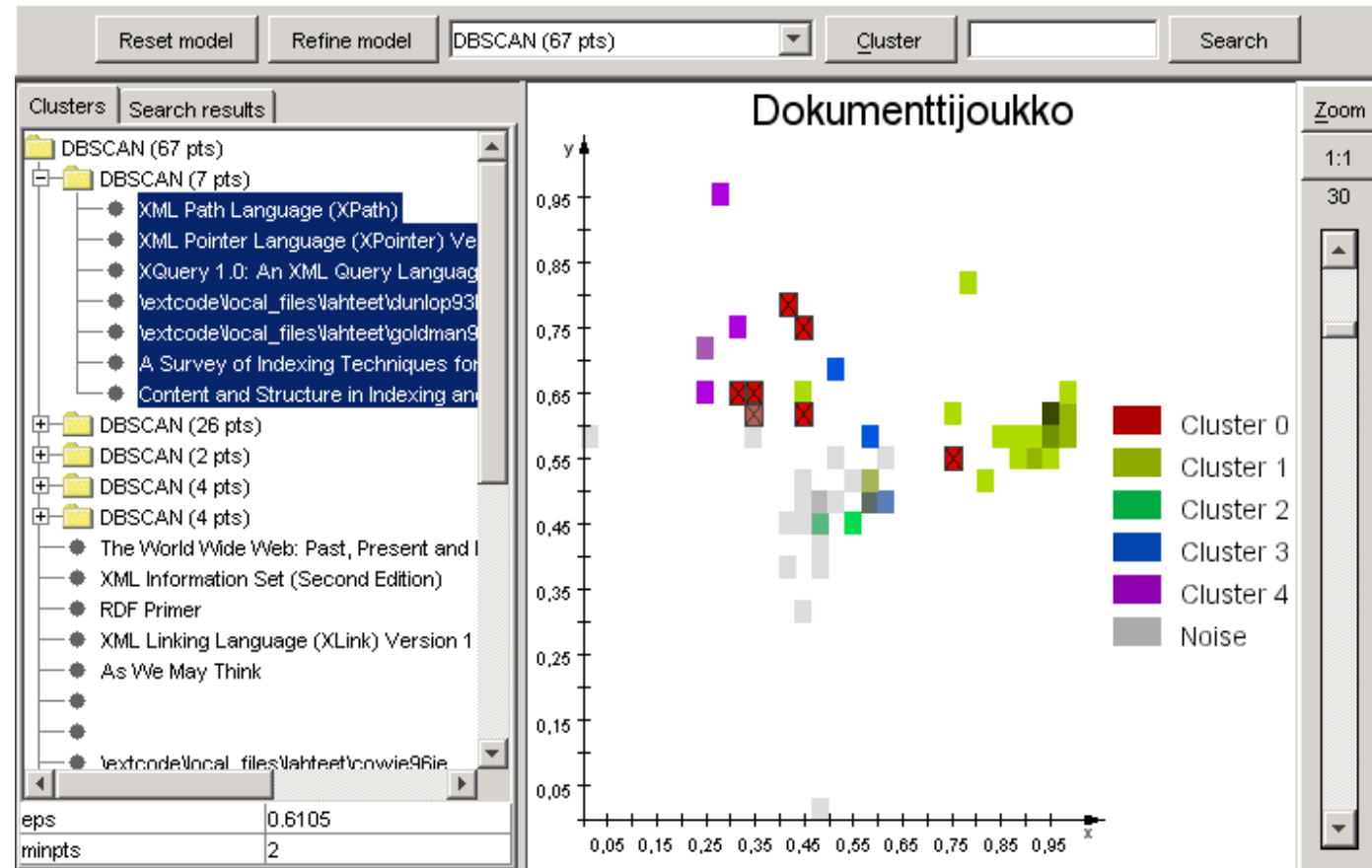
Systems Systems Evolution of automation

## Case 3: References collection

- ExtMiner was used for organizing a collection of references for one of the authors' thesis (title in English: Data Mining from Structured Documents)
- The collection consisted of 145 HTML and PDF documents, the latter were converted to HTML as well. Documents were preprocessed and converted to XML with HTML Tidy.
- Over 50% of the documents did not pass the preprocessing stage (malformed HTML, PDF files that were essentially scanned pictures etc), resulting in 69 indexable documents
- Only index term and header fields were used
- Documents were clustered with both DBSCAN and Group Average hierarchical clustering, resulting in roughly similar cluster models with comparable subject areas

# Case 3: DBSCAN results

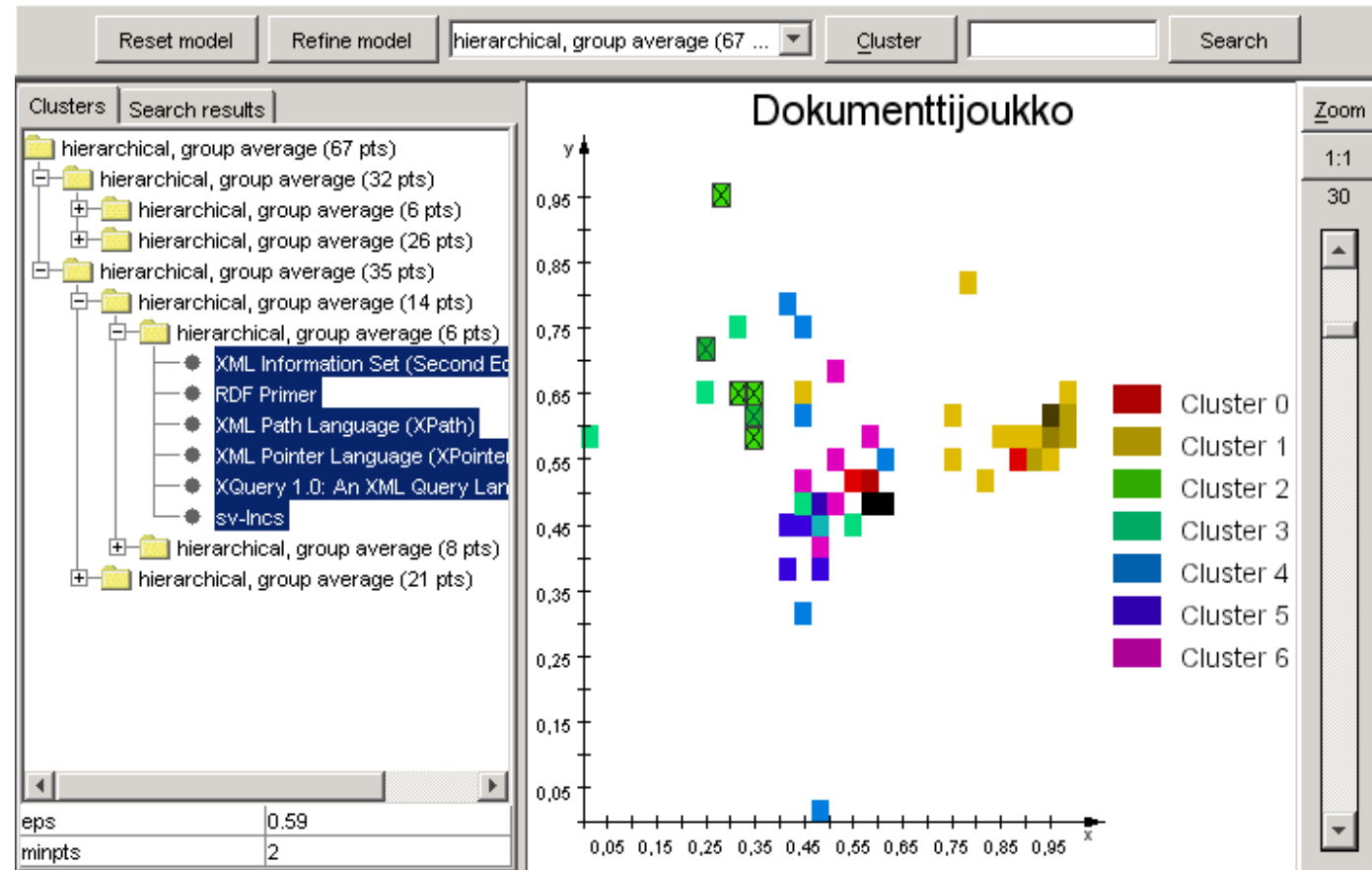
- 5 subject areas + 24 "noise" documents:
  - Generic XML cluster
  - "Main" cluster (IR and document clustering articles)
  - LSI cluster
  - Data mining cluster
  - XML indexing cluster
- DBSCAN parameters were adjusted manually





# Case 3: Group average results

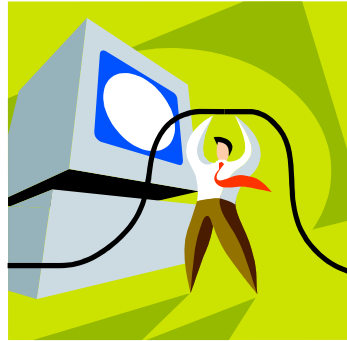
- 2 new subject areas, one dropped, no "noise"
  - Link cluster
  - "General" nontechnical articles (classified as noise by DBSCAN)
    - No LSI cluster
- Hierarchical tree pruning was done manually



# Further research

- ExtMiner shows potential to become a supporting tool for information management in SMEs or organizational workgroups
- Can be used as a platform for further IR or data mining research
- User interface needs further development, currently not suitable for novice users
- Use of ExtMiner requires manual work for preprocessing heterogeneous source documents
- The system should be enhanced with validation functionality for evaluating search and clustering quality with standard test collections
- Manual selection of clustering parameters, hierarchical tree pruning or field weights requires expertise
- Clustering performance was not adequate with large ( $>1000$ ) document collections because of  $O(n^2)$  time complexity (document-to-document similarities).

# Thank you!



[minurmin@cc.jyu.fi](mailto:minurmin@cc.jyu.fi)  
<http://www.mit.jyu.fi/minurmin/>  
<http://extminer.sf.net/>