Querying Bi-level Information

Sudarshan Murthy, David Maier, Lois Delcambre

Department of CSE, OGI School of Science & Engineering at OHSU

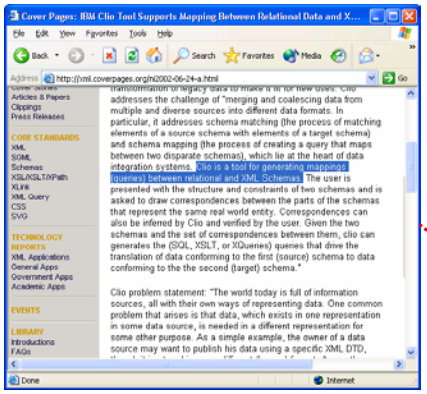
http://cse.ogi.edu/sparce

mailto:smurthy@cse.ogi.edu

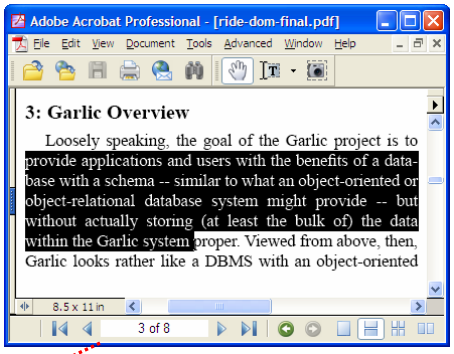
*Picture courtesy of Prof. James Pankow, Dept. of EBS, OGI

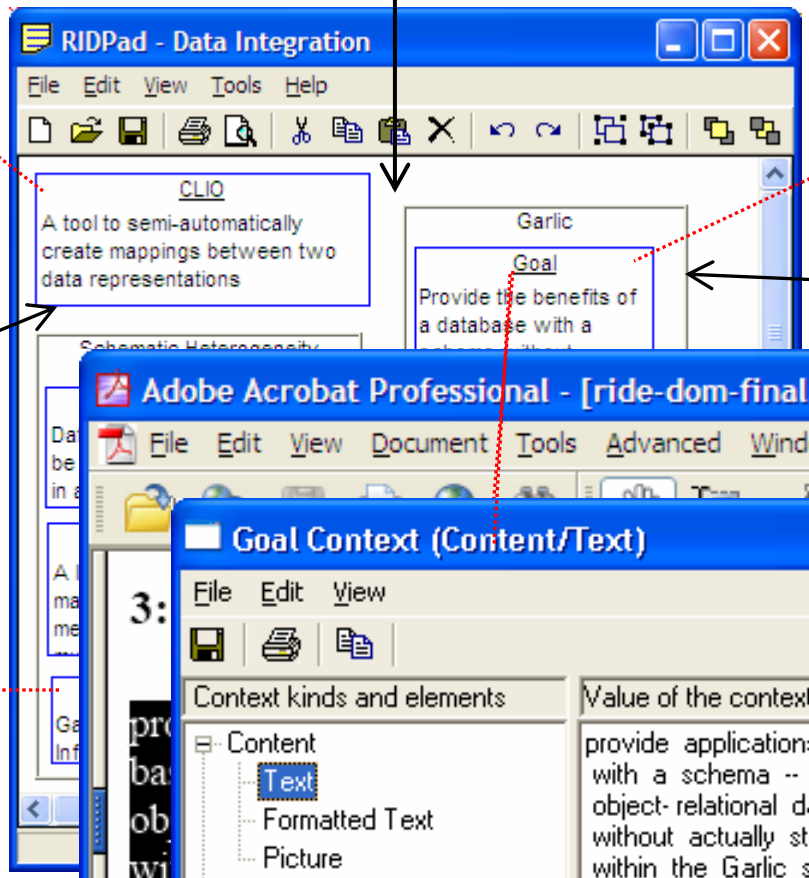
Superimposed Information*

- People often *superimpose* new information onto existing information
  - Annotations, summaries, …
- They use many means
  - Mark up paper
  - Place sticky notes on the paper
- They combine existing information and their interpretations to get "their" view



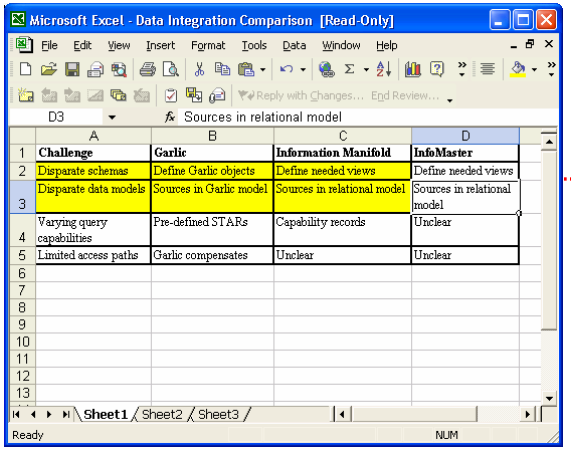
Superimposed information



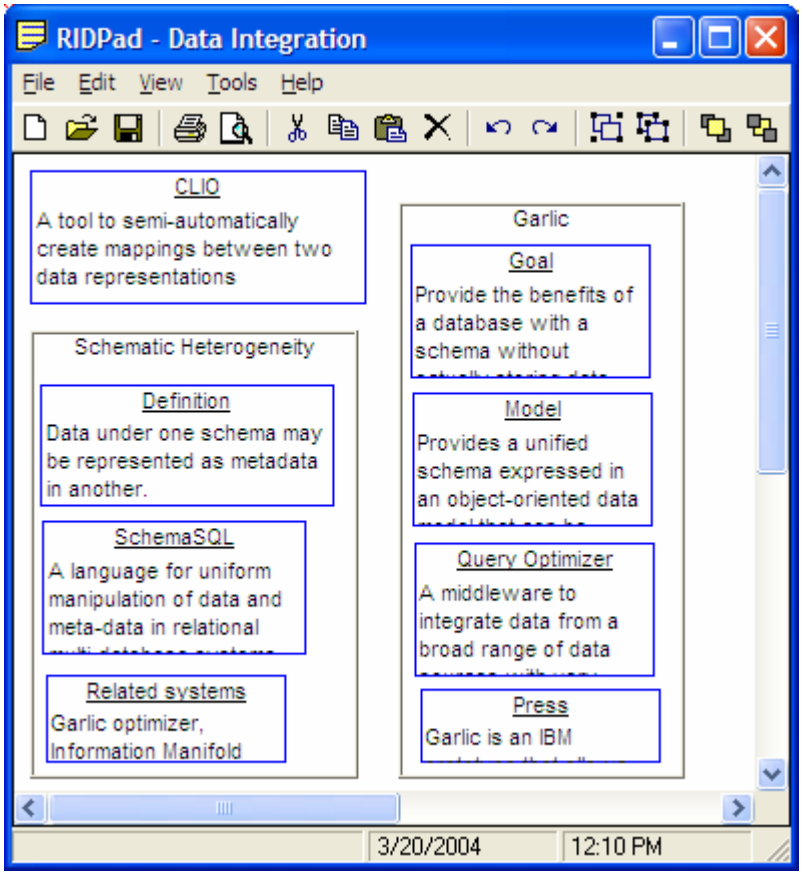


Group

Item



Some Possible Queries



- Show section headings for Garlic items
- List documents consulted
- Create an HTML table of contents from selections

Bi-level queries operate on superimposed information *and* base information

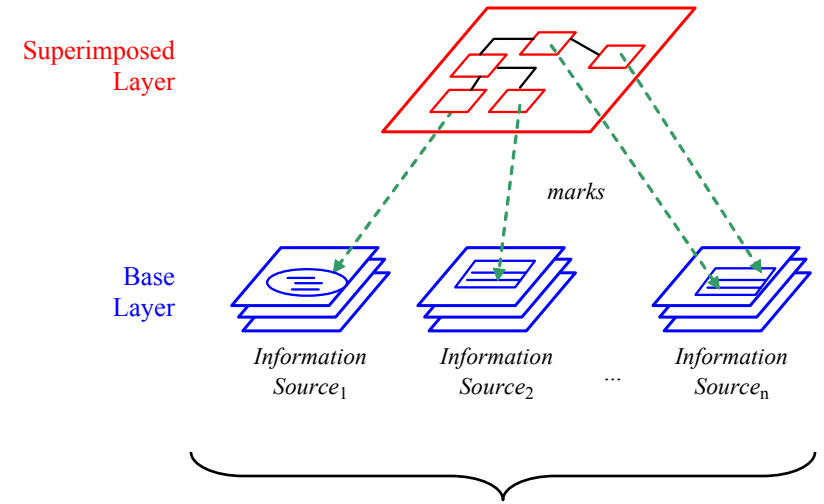
Goal

Query superimposed information and heterogeneous base information of varying granularity, with minimal amount of mediated base information

Outline

- Motivation
- Background
  - Superimposed information management, SPARCE
- Bi-level query system
  - An example implementation
- Discussion
- Conclusion

Superimposing Information

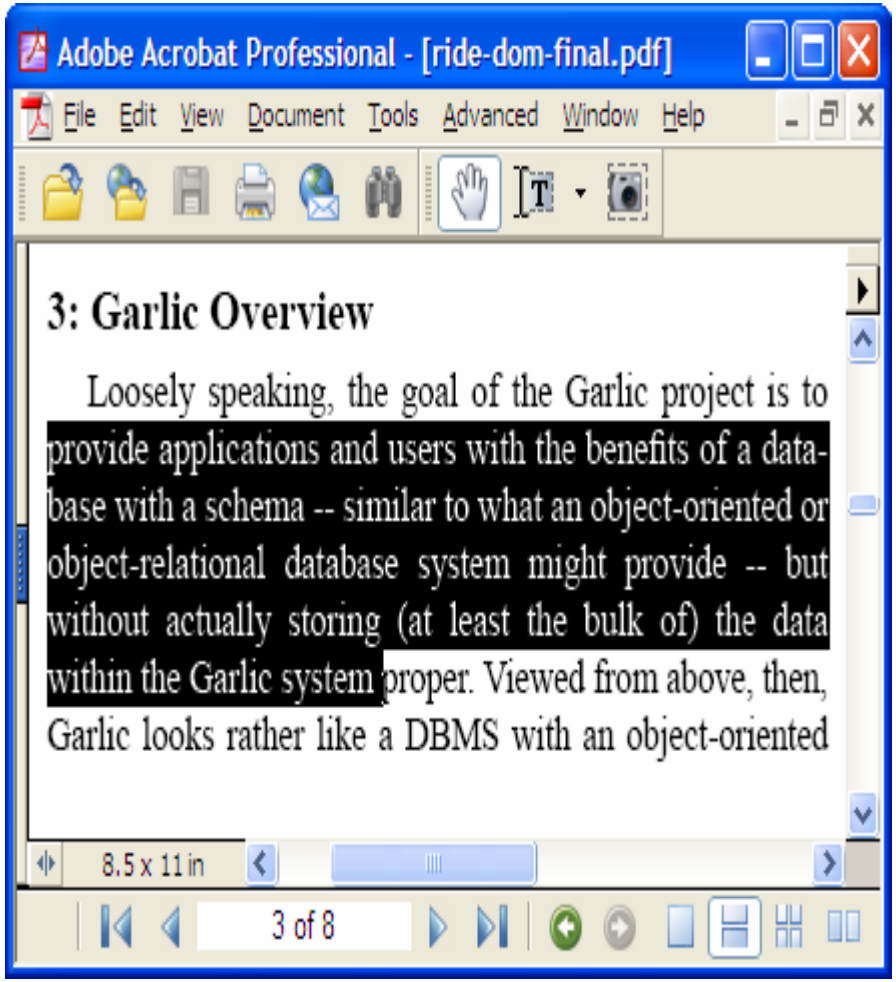


- Overlaying new information on top of existing information
  - Add new data
  - Impose new schema or model
- *Mark* is a reference to base element
  - Many implementations, ~ one per base type
  - Addressing scheme depends on base type

Beyond Browsing

- Marks facilitate browsing
- Bi-level querying requires access to base information
  - *Context* provides access to base information

Excerpts and Contexts



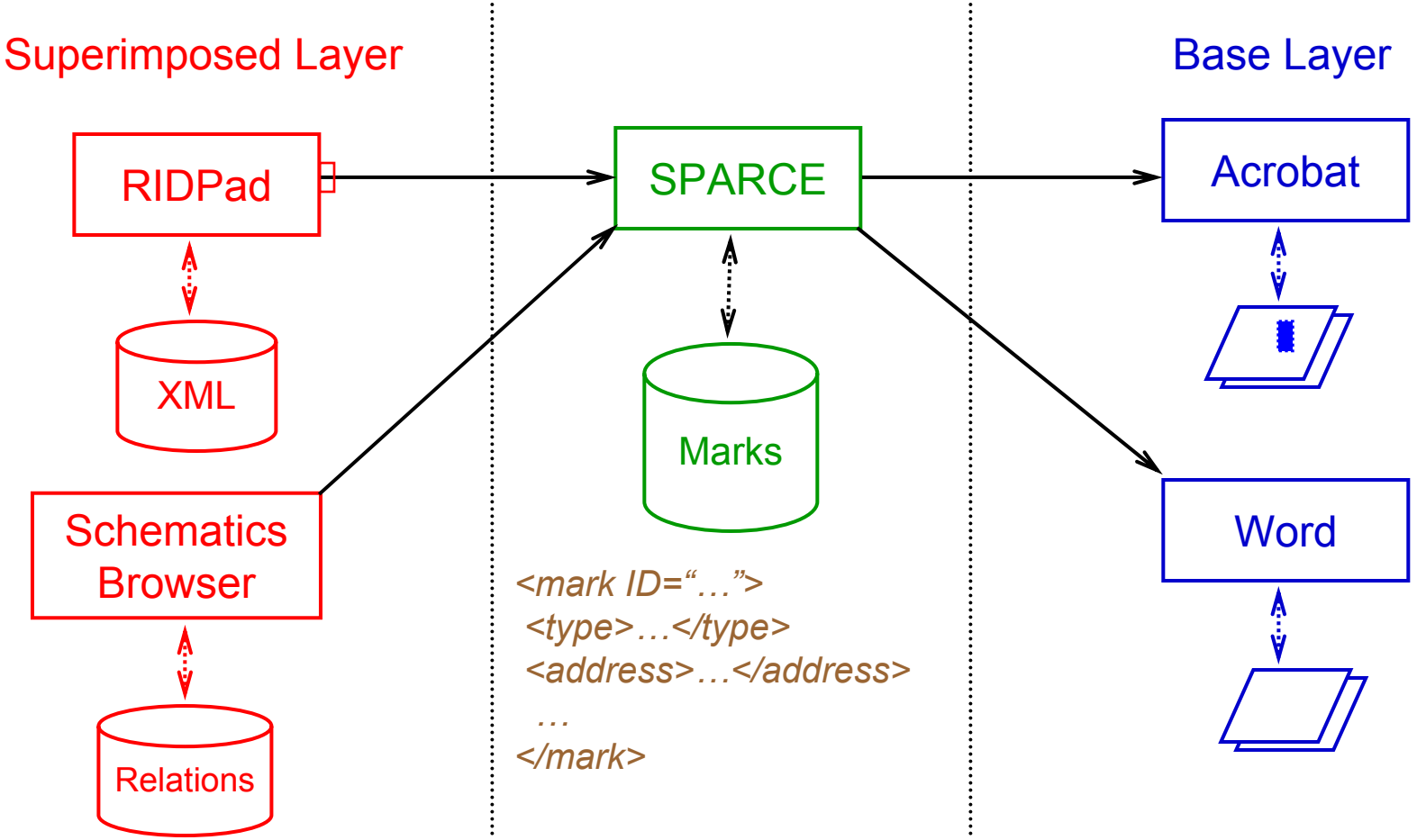
Name	Value
Excerpt	provide applications and users with the benefits of a database with a schema -- similar to … within the Garlic system
Font name	Times New Roman
Section Heading	Garlic Overview

- *Excerpt* is the content of a marked region
- *Context element* is one piece of context
- What constitutes a context varies
- A mediator called *context agent* retrieves context of a mark

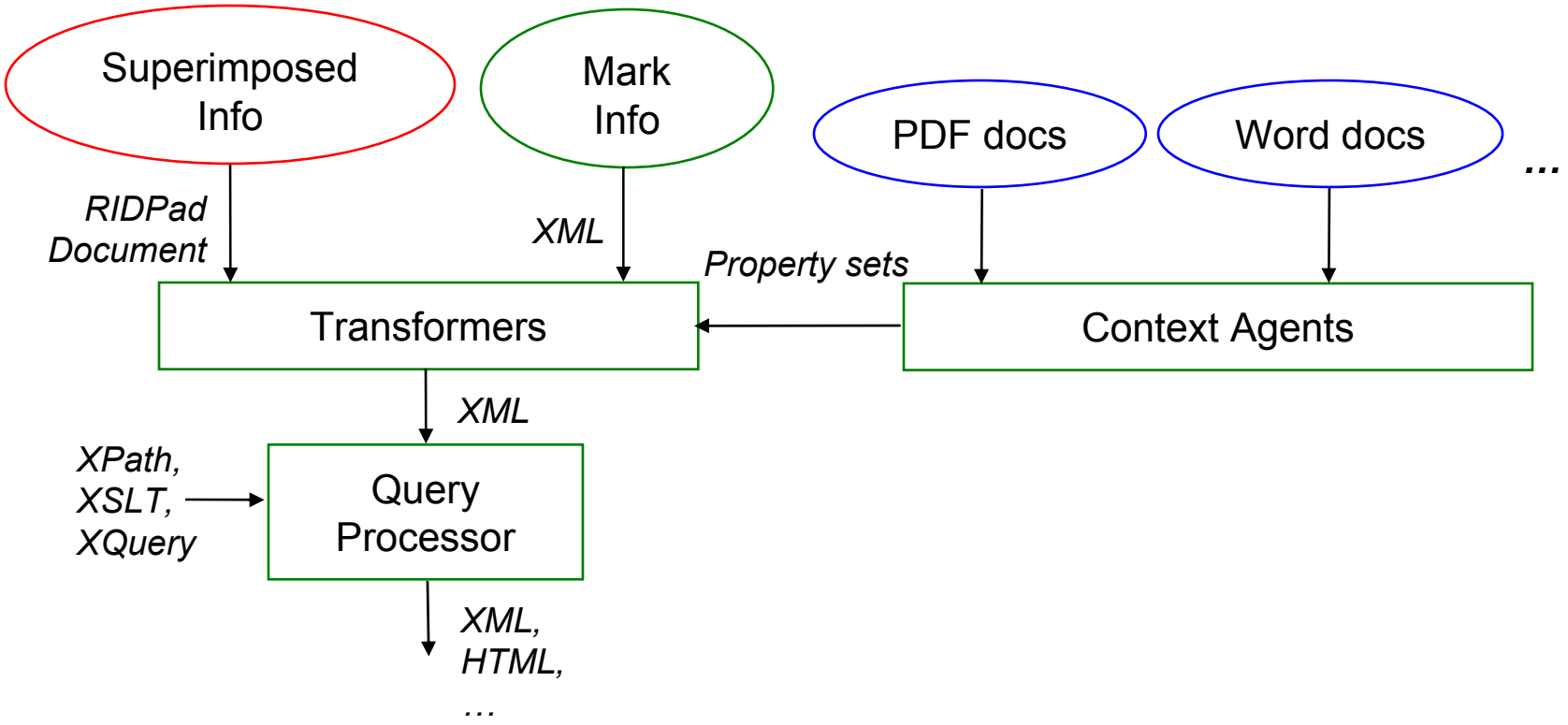
SPARCE

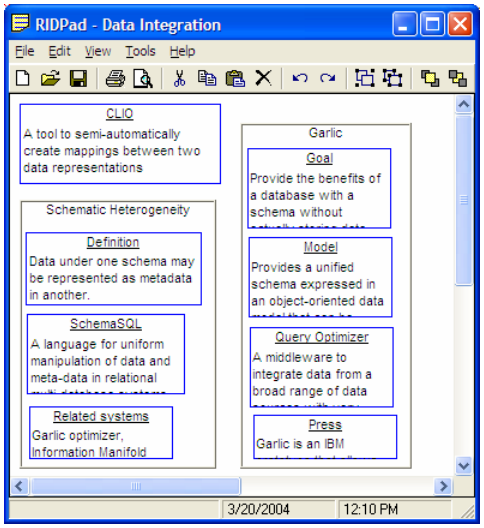
- The *Superimposed Pluggable Architecture for Contexts and Excerpts*
  - Middleware for superimposed information management
  - Provides mark and context management services
- Superimposed applications use SPARCE to activate marks and retrieve context

SPARCE Overview



A Naïve Bi-level Query System





XML Data Generated

```
<?xml version="1.0" ?>
- <RIDPadDocument name="Data Integration">
  - <Group name="Garlic" index="1" left="2955" top="360">
    + <Item name="Press" index="6" left="3000" top="3990">
    - <Item name="Goal" index="4" left="2970" top="675">
      + <Mark ID="AcrobatPDFTextMark20040320105004SURYASMurthy">
      + <Container ID="CClassesCSE606INIride-dom-finalpdf">
      + <Application ID="Acrobat5">
      + <Context objecttype="Mark"
          objectid="AcrobatPDFTextMark20040320105004SURYASMurthy">
        <![CDATA[ Provide the benefits of a database with a schema
        without actually storing data within the Garlic system proper. ]]>
      </Item>
    + <Item name="Query Optimizer" index="3" left="3000" top="2910">
    + <Item name="Model" index="2" left="2985" top="1785">
    </Group>
  + <Group name="Schematic Heterogeneity" index="7" left="120" top="1320">
  + <Item name="CLIO" index="13" left="120" top="120">
  </RIDPadDocument>
```

RIDPAD Document

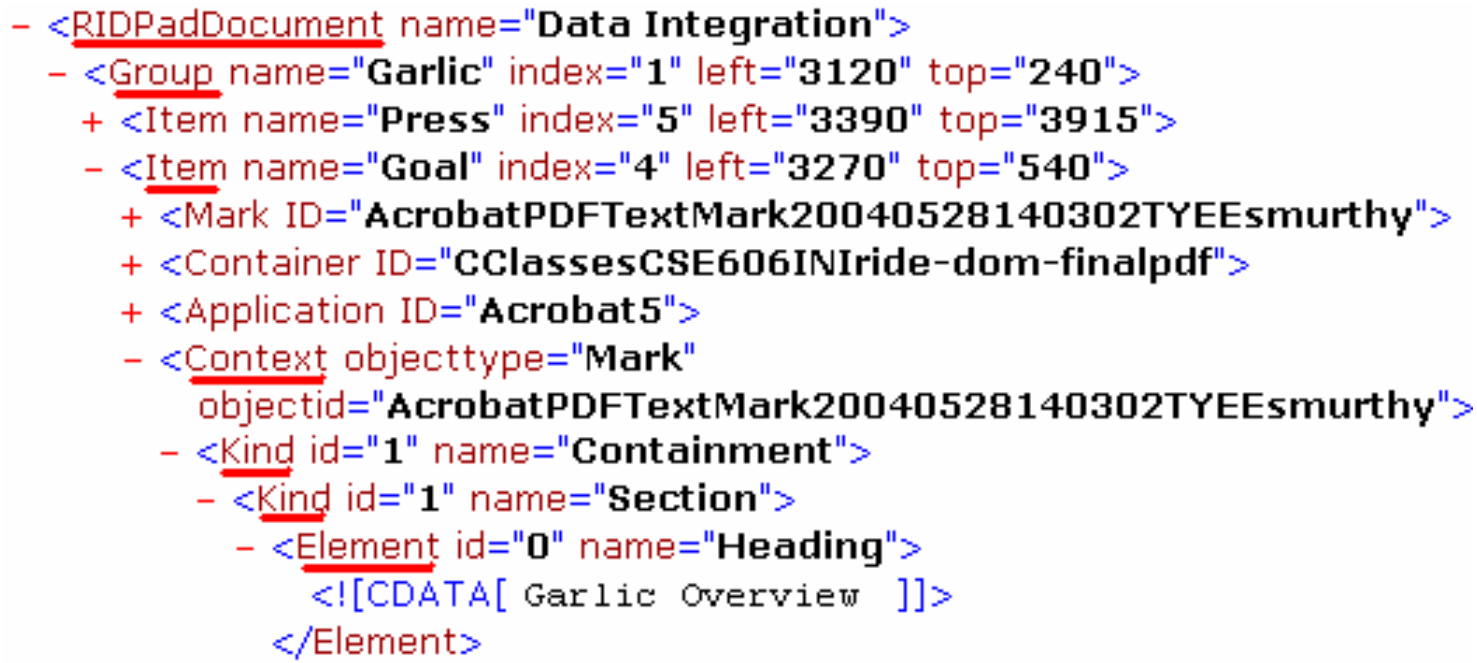
Group

Item

Mark, Container, Application, Context

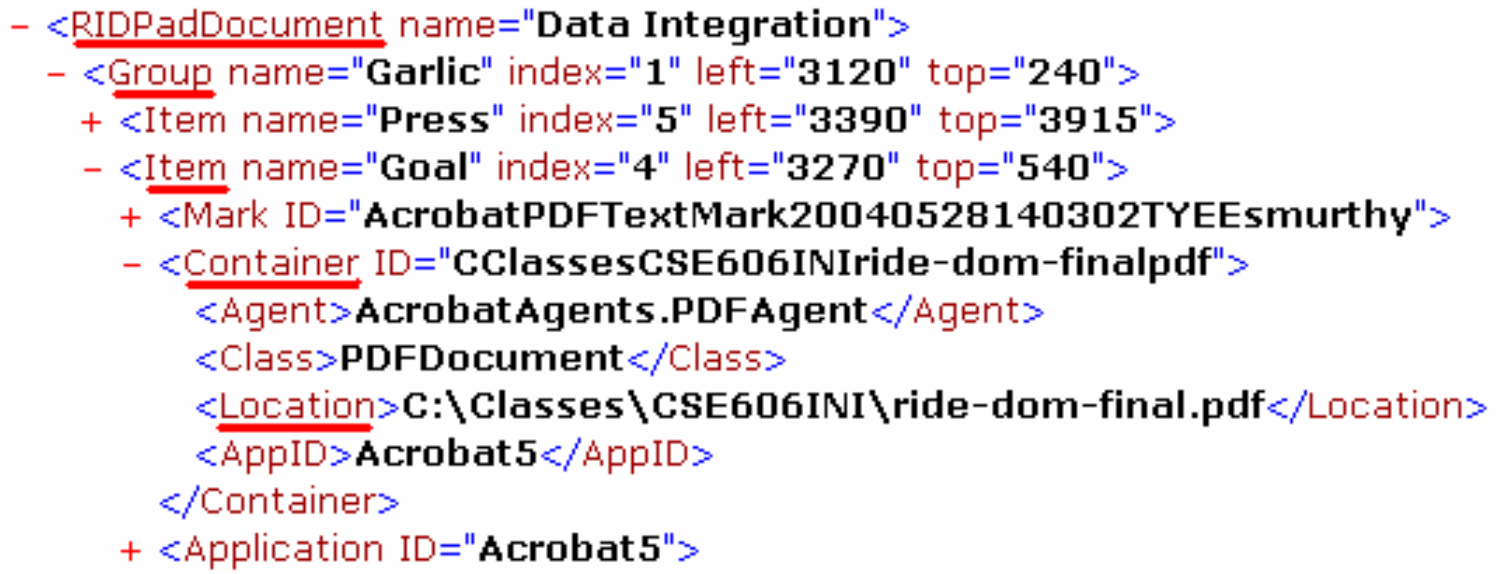
Show Section Headings for Garlic Items

```
//Group[@name='Garlic']/Item/Context/
Kind[@name='Containment']/Kind[@name='Section']/
Element[@name='Heading']
```

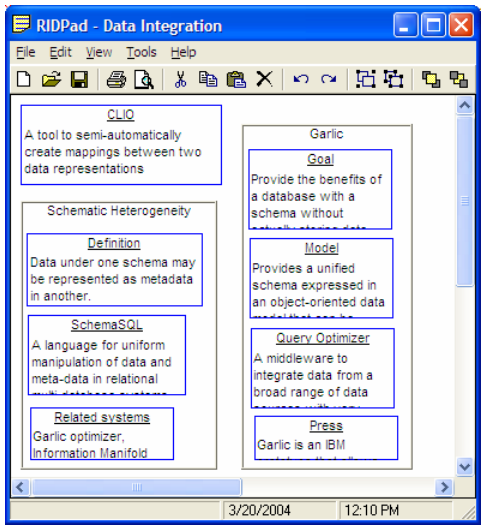


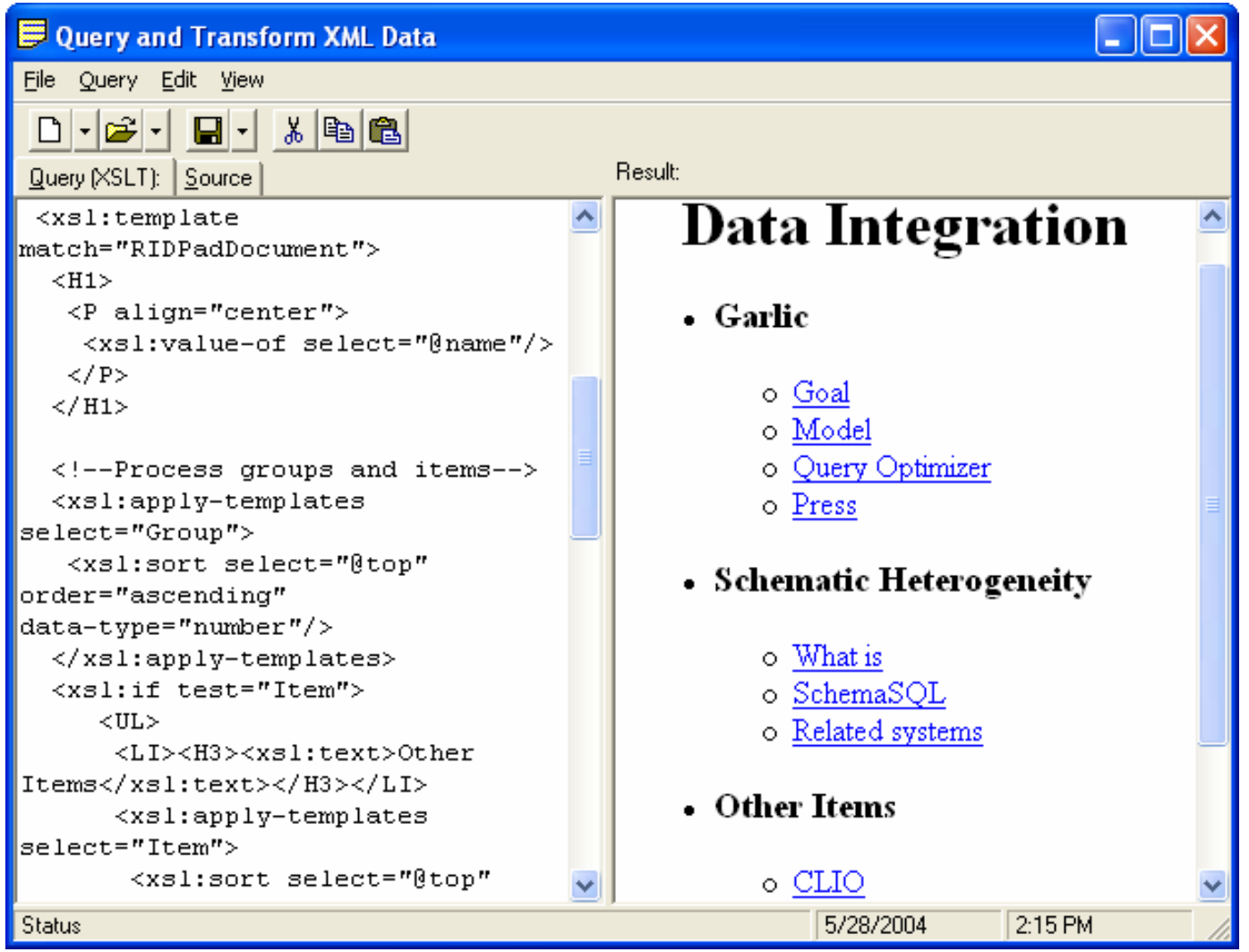
List Documents Selected

```
<Paths> {FOR $l IN
   document("src")//Item/Container/Location
  RETURN <Path> {$l/text()} </Path>
} </Paths>
```



Create HTML Table of Contents

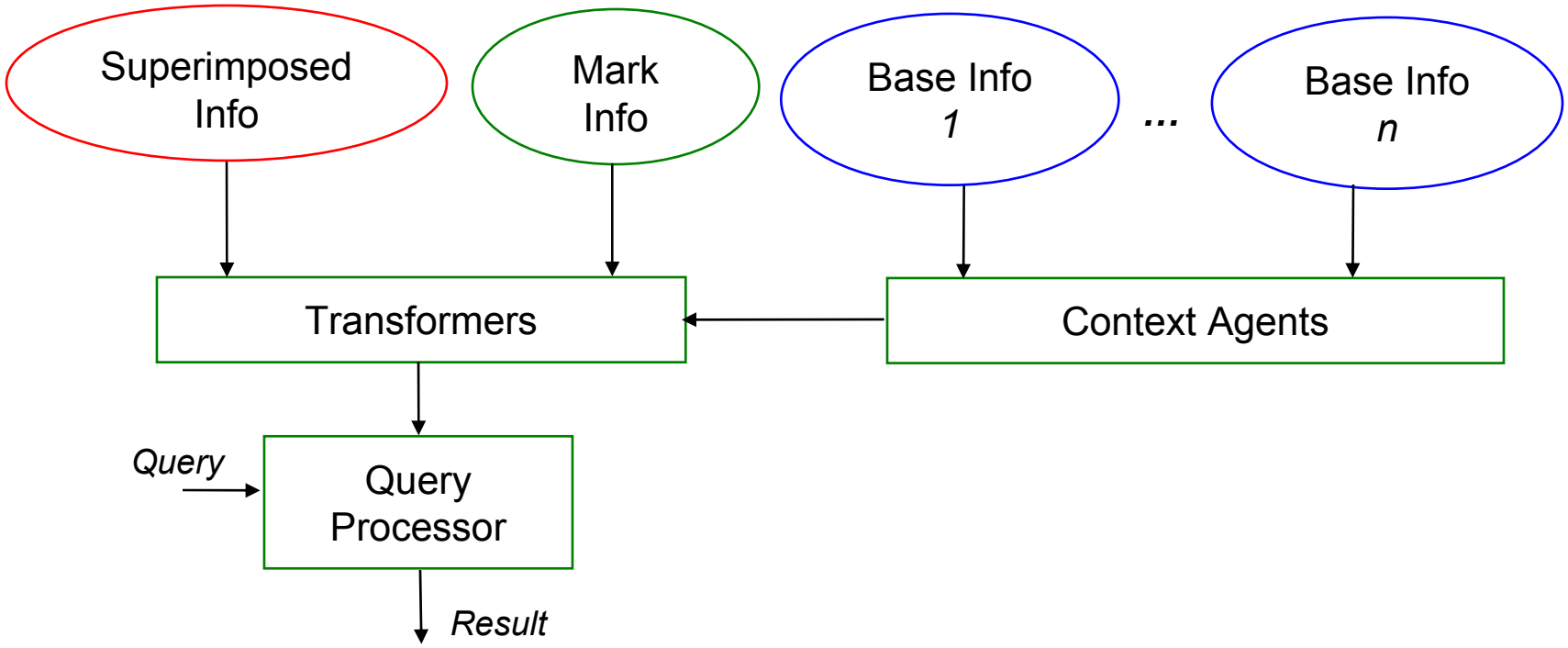




Discussion and Future Work

Query Language and Data Model

- XQuery, etc. may not be appropriate for end users
  - XML may not even be the best data model



Preserve the Layers

<Group name='…'>
<Item name='…' **markid='…'**>
</Item>
</Group>

<Mark id='…'>…</Mark>
<Mark id='…'>…</Mark>

Marks repository

<Context …>…</Context>
<Context …>…</Context>

Build dynamically

Why Preserve the Layers

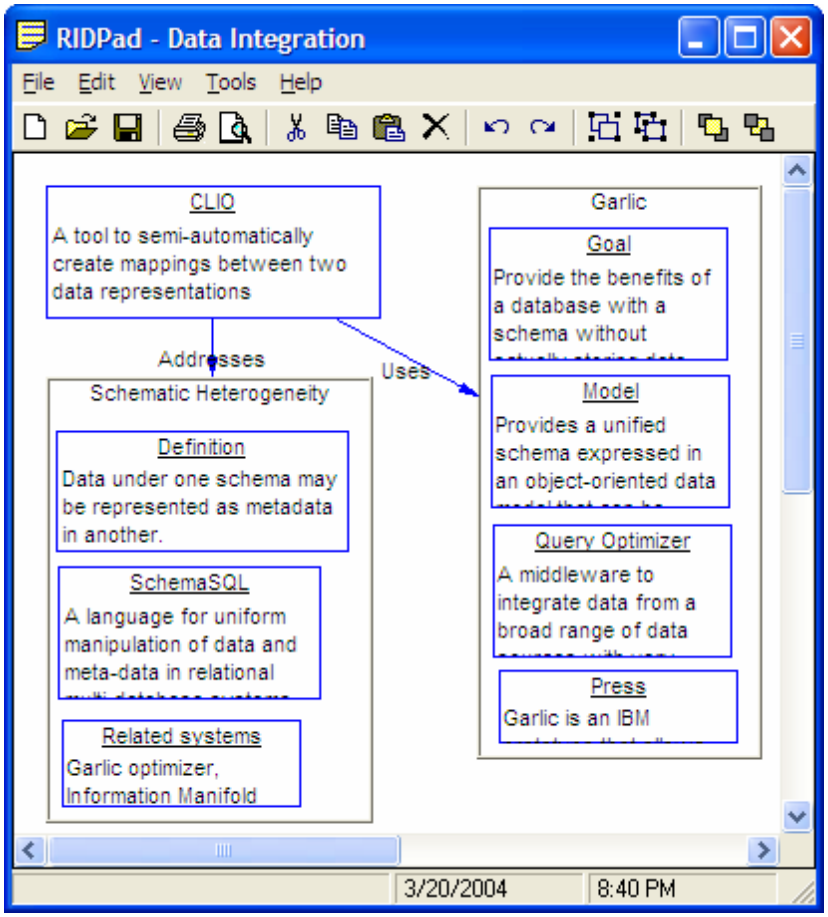
- The information sources are different
  - SI: Superimposed application
  - Marks: SPARCE
  - Contexts: Base applications (via context agents)
- Building a complete hierarchy is unnecessary, and could be inefficient
  - Mark and context information could be replicated
  - Context can be large (broad)
- Joins can provide the same result
  - Views could be defined to “merge” the layers

Add Smarts

*Show **section headings** for **Garlic** items*

- Minimize amount of information retrieved
  - Only some superimposed information elements and marks might qualify, only some context elements might be needed
  - Push ‘selects’ down
- Pass-through queries
  - Some base applications may have query capability
- Explore parallel and distributed query execution

Exploit Relationships



- Relationships in the superimposed layer may help answer new queries
  - What systems does CLIO use?
  - How is CLIO related to SchemaSQL?
- Issues
  - Some relationships may be multi-way; XML is hierarchical

Conclusion

- Enhancing base information with superimposed information makes possible new queries over base information
- Superimposed information and heterogeneous base information may be queried as one
- We have implemented a naïve bi-level query system, but we have many design factors to consider

Questions?

Contact me for a demo

Visit
http://www.cse.ogi.edu/sparce

Related Work

- Garlic: Carey and others, 1995
- MIX: Baru and others, 1999

- MetaXPath: Dyreson and others, 2001
- CXPath: Camillo and others, 2003

- Dexter: Halasz and Schwartz, 1994
- OLE Compound Documents: Microsoft, 1995
- Multivalent Documents: Phelps and Wilensky, 2000