Tagging XML data for our Mediator

Team Members:

Axel Arditi

Steve Beitzel

Eric Jensen

Ali Alhamed

Kalyan Chakravarthy

Valentin John Torres

Team Leader: Michael Saelee

Faculty Project Manager: David Grossman

Overview

- Introduction and Background
- Description of our XML Tagger
- Testing
- Demonstration

Background

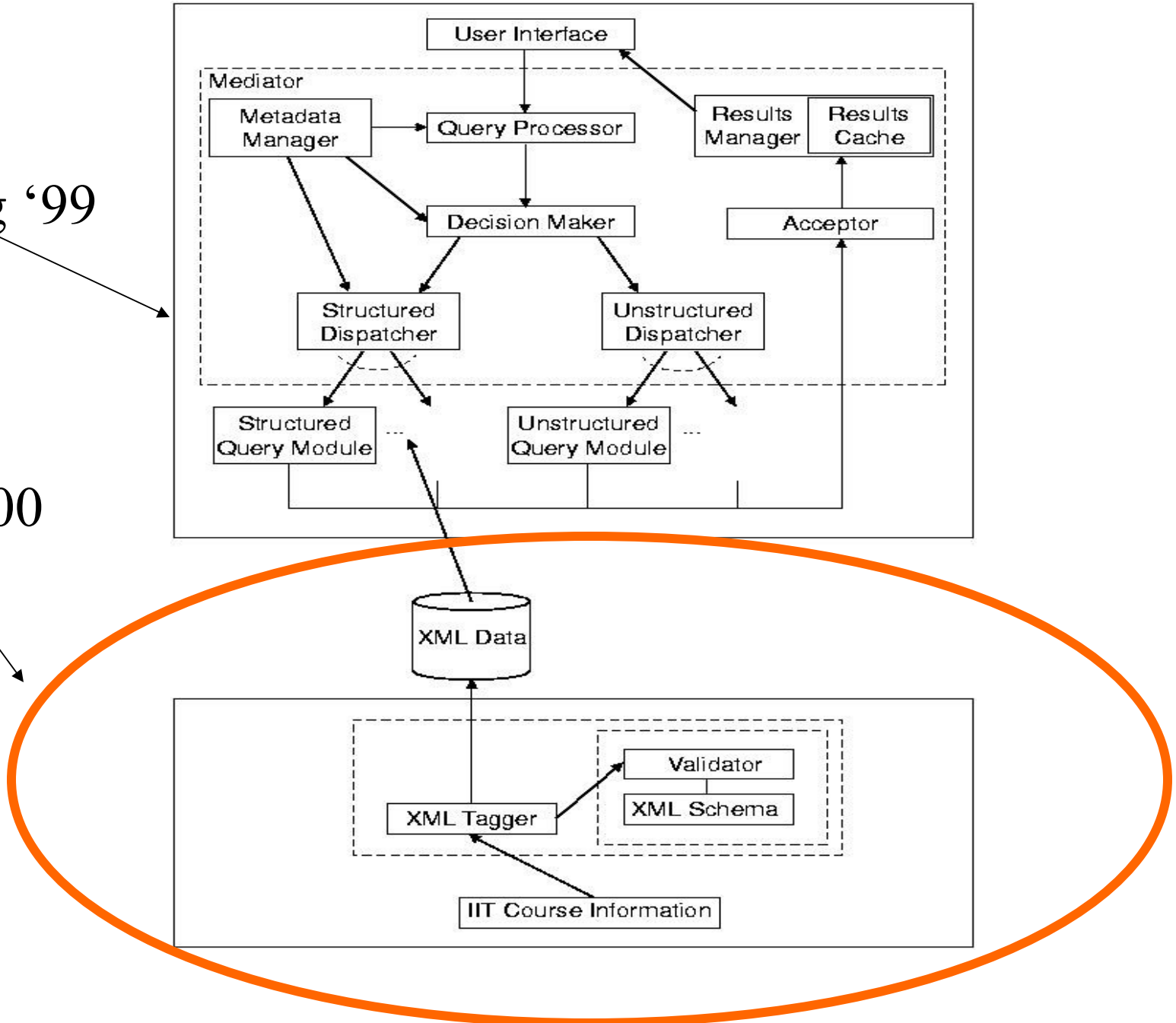
- Last semester we built a prototype mediator which takes a user query and poses it to a variety of different data sources.
- In the worst case, it is as good as existing metasearch engines.
- Example query: "What are the three best restaraunts in Chicago?"
  - Metasearch would search for the word "three"
  - Mediator would identify appropriate sources to answer the question, such as a database of restaurant information.

Adding Data

- We wished to expand the searchable dataset for our mediator.
- We acquired the source text for the Undergraduate Bulletin and wished to add it to our mediator as a semi-structured data source.
- To do this, we needed to build an XML “tagger” that would take a raw text file and add XML “tags” to it, providing it with some structure.
- The tagging process was able to unify the format of all the source data to the Bulletin
- Once data was tagged, we added it to the mediator.

Spring ‘99

Fall ‘00



Input Data

- Three key sections
  - Overview of department with faculty listing
  - Requirements for a major
  - Sample Curriculum

Input Data Section 1:

Computer Science

Computers have changed what we do and how we do it—in our homes, in our offices, and throughout our world. The discipline of computer science focuses upon the many challenging problems encountered in the development and use of computers and computer software. Areas of study in computer science range from theoretical analyses into the nature of computing and computing algorithms, through the development of advanced computing devices and computer networks, to the design and implementation of sophisticated software systems.

The field of applied mathematics explores those branches of mathematics that form the foundation of science and engineering—probability and statistics, numerical analysis, and mathematical modeling. Collectively, these branches define an emerging field of study called computational science and engineering, which uses techniques drawn from applied mathematics and computer science to solve problems from various science and engineering disciplines.

Faculty

Interim Chair
Bogdan Korel
228F Stuart Building
Extension 75150

Professors
Campbell, Carlson, Evens, Frieder

Associate Professors
I. Burnstein, Christopher, Greene, Korel, Robergé

Assistant Professors
Chang, Dickens, Hood, Orlandic, Wan

Research Associate Professor
Elrad

Adjunct Associate Professors
Biernat, Chafi, Drakopoulos, Lidinsky, Soneru

Adjunct Assistant Professors
Nowicki, Trygstad, Woyna

Lecturer
Brandle

Instructors
M. Bauer, Bistriceanu, Manov

Faculty Emeriti
C. Bauer

Input Data Section 2:

- Description……
- Title……
- Required Courses…….

Computer Science

The department offers two undergraduate programs in computer science: a Bachelor of Science in Computer Science and an Applied Science for the Professions Bachelor of Science in Computer Information Systems. Both programs provide an excellent background in computer science and allow for ample study in other areas. Where these programs differ is in the approach they take to computer science. The B.S. in Computer Science provides an in-depth experience focusing on the theory and practice of computer science while the B.S. in Computer Information Systems provides a more interdisciplinary experience, balancing study in computer science with study in another field. In addition to these programs in computer science, the Department of Computer Science and the Department of Electrical and Computer Engineering jointly offer a Bachelor of Science in Computer Engineering. This program focuses on both the digital electronics hardware used in computer systems and the software that controls this hardware, with an emphasis on the design and implementation of computer-controlled systems. This program is described in detail on page 75.

All three programs begin with a set of introductory courses that work together to provide students with a firm foundation in computer science. These introductory courses include weekly labs in which students use state-of-the-art software development techniques (object-oriented programming in C++, for instance) to create solutions to interesting problems. The department's unique four-phase laboratory model encourages student creativity by providing ample opportunity for constructive feedback on each student's efforts. Having completed the introductory core, a student is prepared to work independently within a well-structured design framework—in the classroom or on the job.

The last two years of study build upon this foundation. The Bachelor of Science in Computer Science focuses on the concepts and techniques used in the design and development of advanced software systems. Students in this program explore the conceptual underpinnings of computer science—its fundamental algorithms, programming languages, operating systems, and software engineering techniques. In addition, students choose from a rich set of electives—including computer graphics, artificial intelligence, database systems, computer architecture, and computer networks, among others. As with the introductory sequence, these advanced courses stress "hands-on" learning by doing. A generous allotment of free electives allows students to combine study in computer science with study in another field—either by taking a well-defined specialized minor in another discipline or by working with an adviser to formulate a program that combines experiences across disciplines.

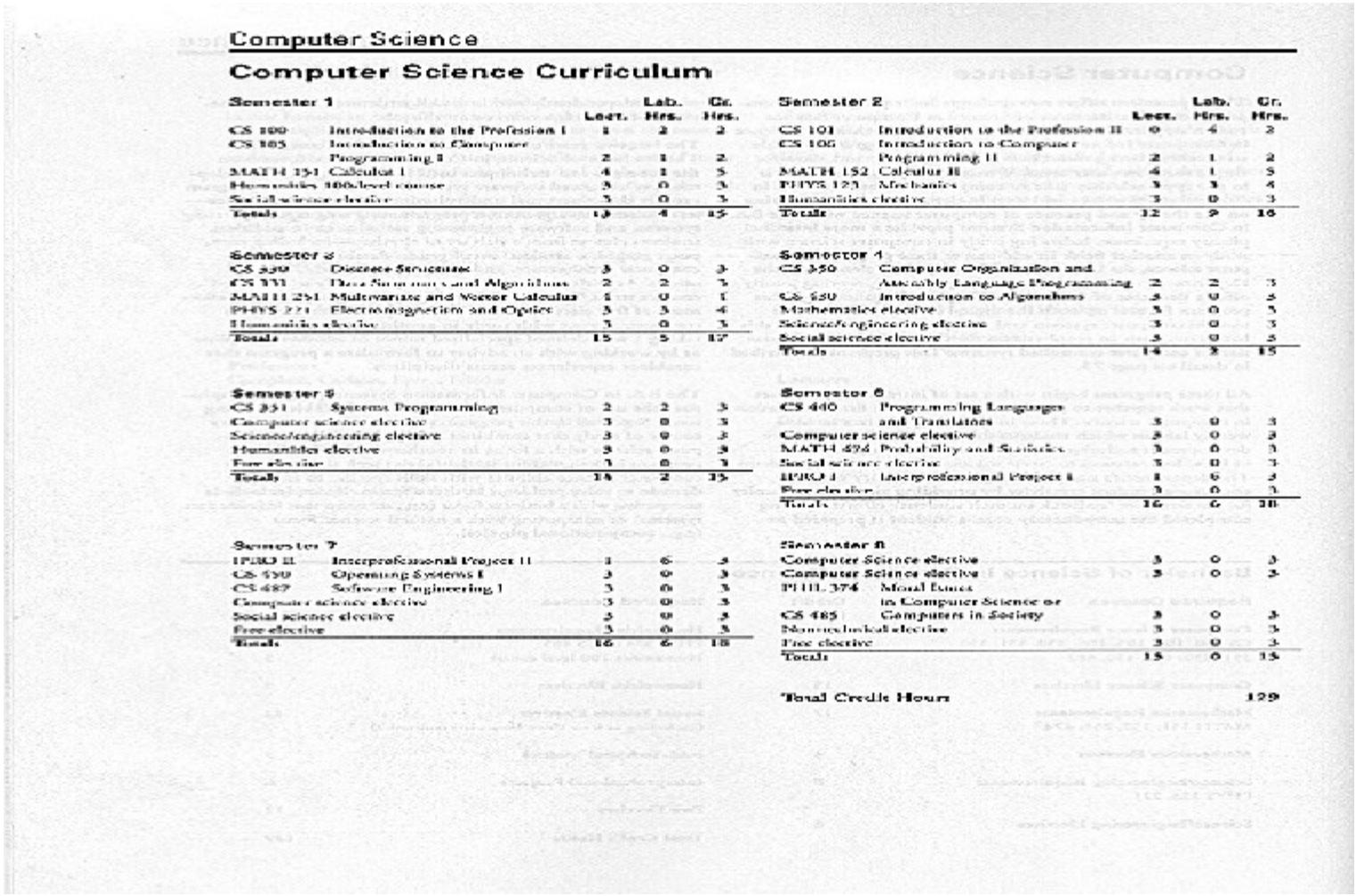
The B.S. in Computer Information Systems program emphasizes the use of computers as sophisticated problem-solving tools. Students in this program pursue an interdisciplinary course of study that combines a solid foundation in computer science with a focus in another discipline. This program is designed for students who seek to blend their computer science abilities with skills specific to another domain to solve problems in that domain. Examples include computing with a business focus (e.g., management information systems) or computing with a natural science focus (e.g., computational physics).

Bachelor of Science in Computer Science

Required Courses	Credit Hours
Computer Science Requirements CS 100, 101, 105, 106, 330, 331, 350, 351, 430, 440, 450, 487	32
Computer Science Electives	15
Mathematics Requirements MATH 151, 152, 251, 474	17
Mathematics Electives	3
Science/Engineering Requirements PHYS 123, 221	8
Science/Engineering Electives	6

Required Courses	Credit Hours
Humanities Requirements PHIL 374 or CS 485	3
Humanities 100-level course	3
Humanities Electives	9
Social Science Electives (including at least three hours in economics)	12
Non-Technical Elective	3
Interprofessional Projects	6
Free Electives	12
Total Credit Hours	129

Input Data Section 3



Computer Science

Computer Science Curriculum

Semester 1		Lect.	Lab. Hrs.	Cr. Hrs.
CS 100	Introduction to the Profession I	1	2	2
CS 105	Introduction to Computer Programming II	2	1	2
MATH 151	Calculus I	4	1	5
Humanities 100-level course		3	0	3
Social science elective		3	0	3
Totals		13	4	15

Semester 2		Lect.	Lab. Hrs.	Cr. Hrs.
CS 101	Introduction to the Profession II	0	4	2
CS 106	Introduction to Computer Programming II	2	1	2
MATH 152	Calculus II	4	1	5
PHYS 123	Mechanics	3	3	4
Humanities elective		3	0	3
Totals		12	9	16

Semester 3		Lect.	Lab. Hrs.	Cr. Hrs.
CS 250	Discrete Structures	3	0	3
CS 251	Data Structures and Algorithms	2	2	3
MATH 251	Multivariate and Vector Calculus	4	0	4
PHYS 271	Electromagnetism and Optics	3	3	4
Humanities elective		3	0	3
Totals		15	5	17

Semester 4		Lect.	Lab. Hrs.	Cr. Hrs.
CS 350	Computer Organization and Assembly Language Programming	2	2	3
CS 351	Introduction to Algorithms	3	0	3
Mathematics elective		3	0	3
Science/engineering elective		3	0	3
Social science elective		3	0	3
Totals		14	2	15

Semester 5		Lect.	Lab. Hrs.	Cr. Hrs.
CS 351	Systems Programming	2	2	3
Computer science elective		3	0	3
Science/engineering elective		3	0	3
Humanities elective		3	0	3
Free elective		3	0	3
Totals		14	2	15

Semester 6		Lect.	Lab. Hrs.	Cr. Hrs.
CS 440	Programming Languages and Translators	3	0	3
Computer science elective		3	0	3
MATH 474	Probability and Statistics	3	0	3
Social science elective		3	0	3
IPRO I	Interprofessional Project I	1	6	3
Free elective		3	0	3
Totals		16	6	18

Semester 7		Lect.	Lab. Hrs.	Cr. Hrs.
IPRO II	Interprofessional Project II	1	6	3
CS 530	Operating Systems I	3	0	3
CS 487	Software Engineering I	3	0	3
Computer science elective		3	0	3
Social science elective		3	0	3
Free elective		3	0	3
Totals		16	6	18

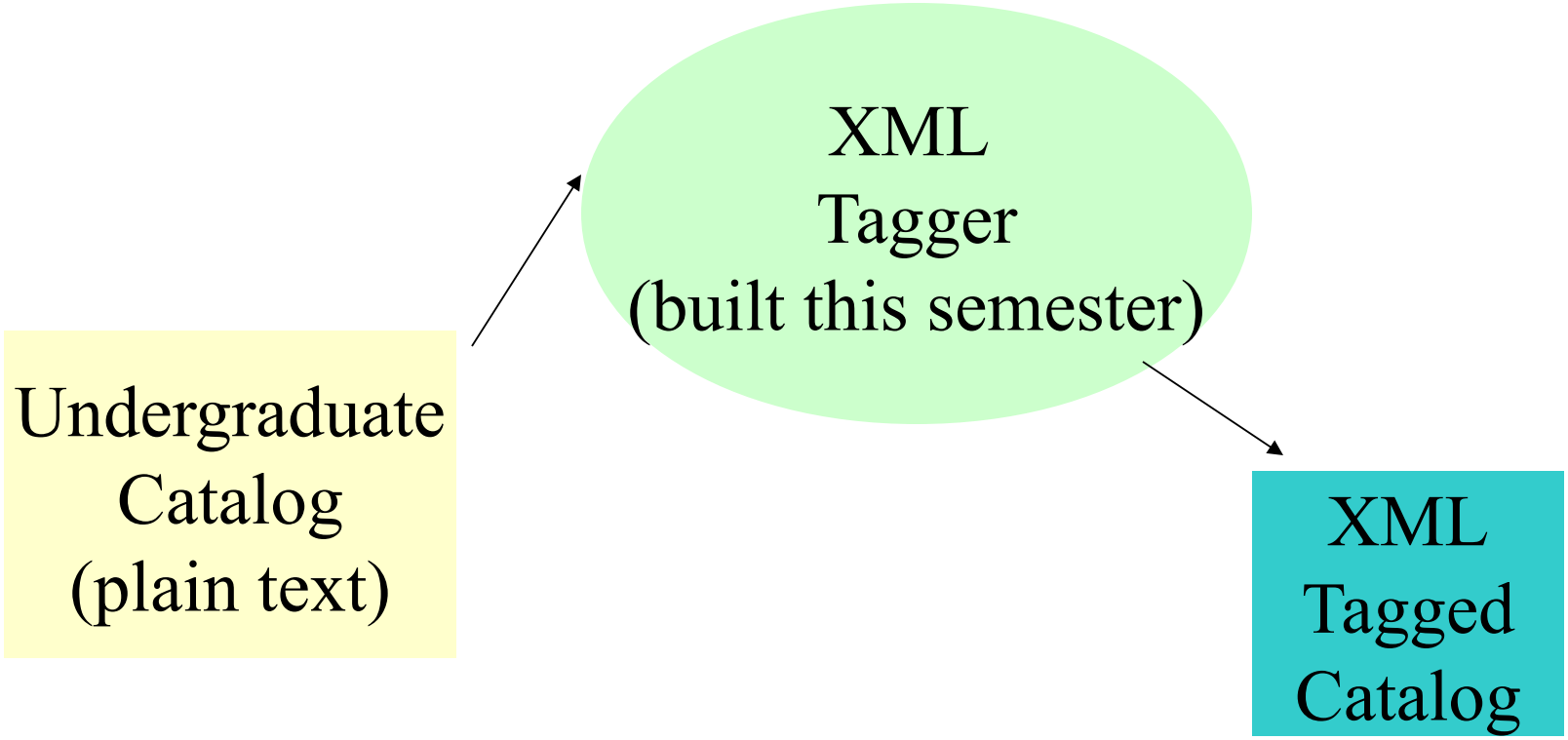
Semester 8		Lect.	Lab. Hrs.	Cr. Hrs.
Computer Science elective		3	0	3
Computer Science elective		3	0	3
PHIL 374	Moral Issues in Computer Science or			
CS 485	Computers in Society	3	0	3
Humanities/social elective		3	0	3
Free elective		3	0	3
Totals		15	0	15

Total Credit Hours 129

XML Tagging Text

- Support for quantitative, structured queries depends on having an element of structure present in the data
- XML, the eXtensible Markup Language is a semi-structured format
- We developed software to add XML tags to the IIT Undergraduate Bulletin, enabling our mediator to answer quantitative queries

Tagging the data



Ready for feeding to the mediator.

Sample XML Output

```
<department xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="/usr/local/share/ipro/IITCourseBulletin.xsd">
   <name>Computer Science</name>
   <overview> Computers have changed what we do and how we do it …</overview>
   <faculty>
      <member chairType="Interim" isChair="true">
         <name> Edward Reingold </name>
         <office>
            <number>236B</number>
            <building>Stuart Building</building>
         </office>
         <extension>Extension 75150</extension>
      </member>
      <member rank="Professor">
         <name>Campbell</name>
      </member>
      <member rank="Professor">
         <name>Carlson</name>
      </member>
       ….
```

Sample XML Query

- To obtain a listing of "member" elements in a specific department's faculty section for full professors
  - Natural language query: "*find all full professors in the computer science department*"
  - XQL query: //department[name="Computer Science"]
    //faculty/member[@rank="Professor"]

Result:

```
<xql:result xmlns:xql="http://metalab.unc.edu/xql/">
  <member rank="Professor">
    <name>Evens</name>
  </member>
  <member rank="Professor">
    <name>Frieder</name>
  </member>
</xql:result>
```

Approach

- Documented existing mediator
- Learned Java
- Worked to incorporate IIT Undergraduate Bulletin Data into our mediator
  - Learned XML
  - Designed XML schema
  - Partitioned input documents into three segments (one for each developer).

Team Meetings

- Extensive code review of all software
- Agreed to coding standards and modifications to the schema
- Take minutes and publish on web site http://cs.iit.edu/~wsearch
- Develop schedule and check schedule at each meeting.

Results

- 1100 lines of java that has produced 26 files of correctly tagged XML
- Simple integration of XML data into the mediator.
- Simple user interface to our newly tagged XML documents

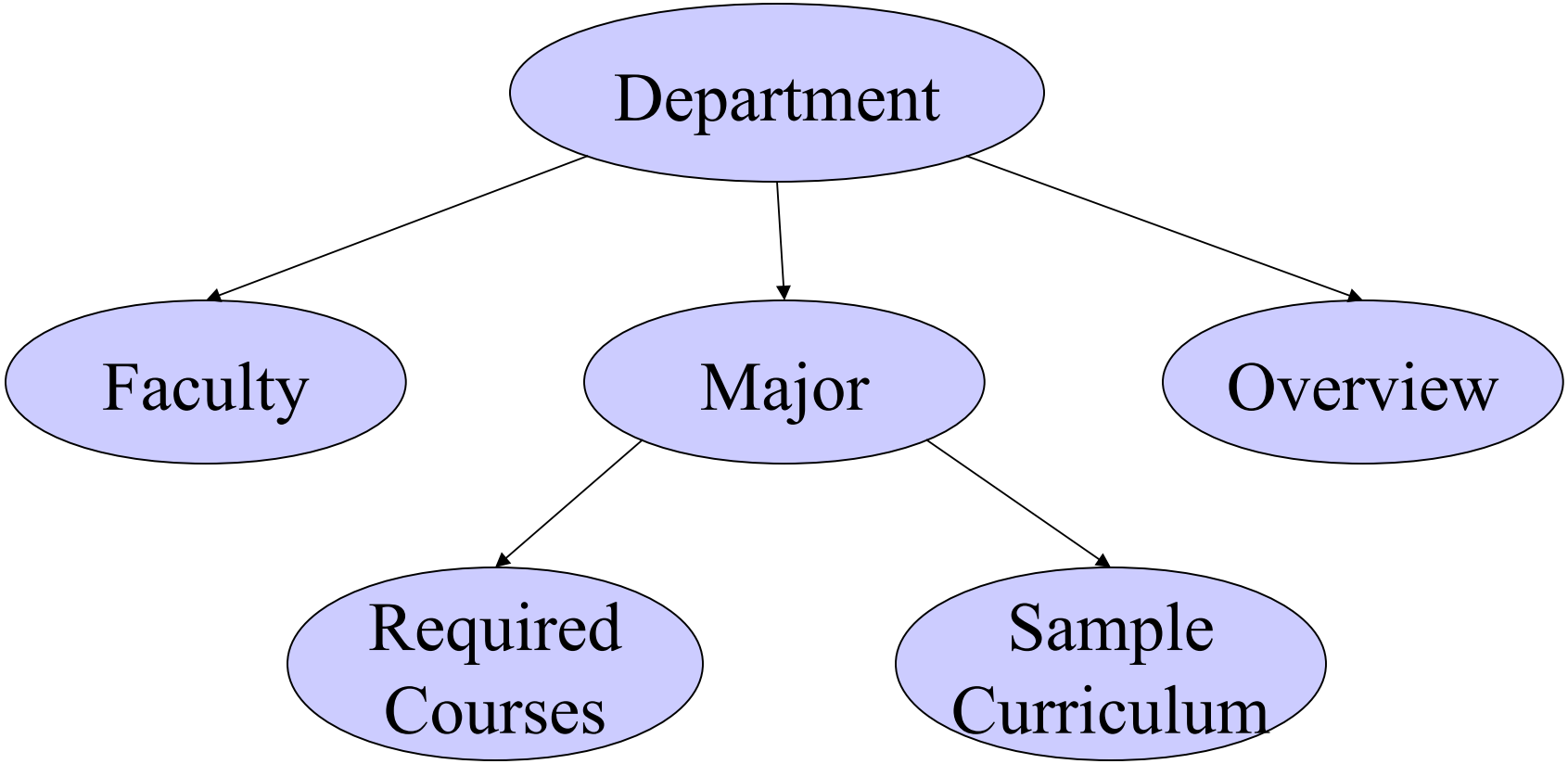
Team Assignments

- Steve and Eric
  - maintained XML schema ensured all built components talked to each other. Built configuration file.
- Axel
  - Overview, department title, faculty
- Kalyan
  - Major, description, required courses
- Ali
  - Sample Curriculum
- Val
  - independent software testing
  - provided non-CS insight into what we were doing

XML Schema

- The first step in tagging the bulletin was to define what XML we wanted as an output
- We used the XML Schema language to precisely define the format that each of the department entries from the bulletin should take
- We used this schema to validate that we produced the correct XML for the data

High Level Schema



Sample from our Schema

```
<schema>
 <annotation>
  <documentation>
   Illinois Institute of Technology course bulletin schema
   (undergraduate, but possibly graduate in the future)
  </documentation>
 </annotation>
 <element name="department" type="DepartmentType"/>
 <complexType name="DepartmentType">
  <element name="name" type="string"/>
  <element name="overview" type="string"/>
  <element name="homepage" type="string" minOccurs="0" maxOccurs="1"/>
  <element name="faculty" type="FacultyType"/>
  <element name="major" maxOccurs="unbounded">
   <complexType>
    <element name="name" type="string"/>
    <element name="description" type="string"/>
```

Tagger Generalization

- All document-specific data, including anchor strings and regular expressions, are stored in a simple configuration file.
- This allows us to easily adapt the tagger to support changes in the source data or the XML schema.
- In addition to being flexible, this approach adds extensibility, making it easy for us to add new methods to the parsing process.

Example of Config File

```
section0=departmentName
section1=overview
section2=faculty
section3=major
section4=requiredCourses
section5=description
section6=sampleCurriculum

sectionRE0=\\A([^\n]+)\n
sectionRE1=\\A[^\n]+\n(.+)\n\nFaculty\n
sectionRE2=\\A[^\n]+\n.+\n\nFaculty\n(.+)
sectionRE3=(.+)([0-9]+)

semester=^Semester ([0-9]+)
course=^([A-Z]+) +([0-9]{3}) +([A-Z].*[A-Za-z]) +([0-9]*) +([0-9]*) +([0-9]*)$
ipro=^([A-Z]+) +([I]{1,2}) +([A-Z].*[A-Za-z]) +([0-9]*) +([0-9]*) +([0-9]*)$
elective1=^(.+ [Ee]lective[s]*) +([0-9]*) +([0-9]*) +([0-9]*)$
elective2=^(.+) +([1-7]00)-level course +([0-9]*) +([0-9]*) +([0-9]*)$
elective3=^(.+) elective/minor +([0-9]*) +([0-9]*) +([0-9]*)$
courseName=([A-Z].*[A-Za-z]) +[|] +([A-Z].*[A-Za-z]) \n*
totalHours=^Totals +([0-9]{1,2}*) +([0-9]{1,2}*) +([0-9]{1,2}*)$
totalCurriculumHours=^Total Credit Hours +([0-9]{2,3})$
. . .
```

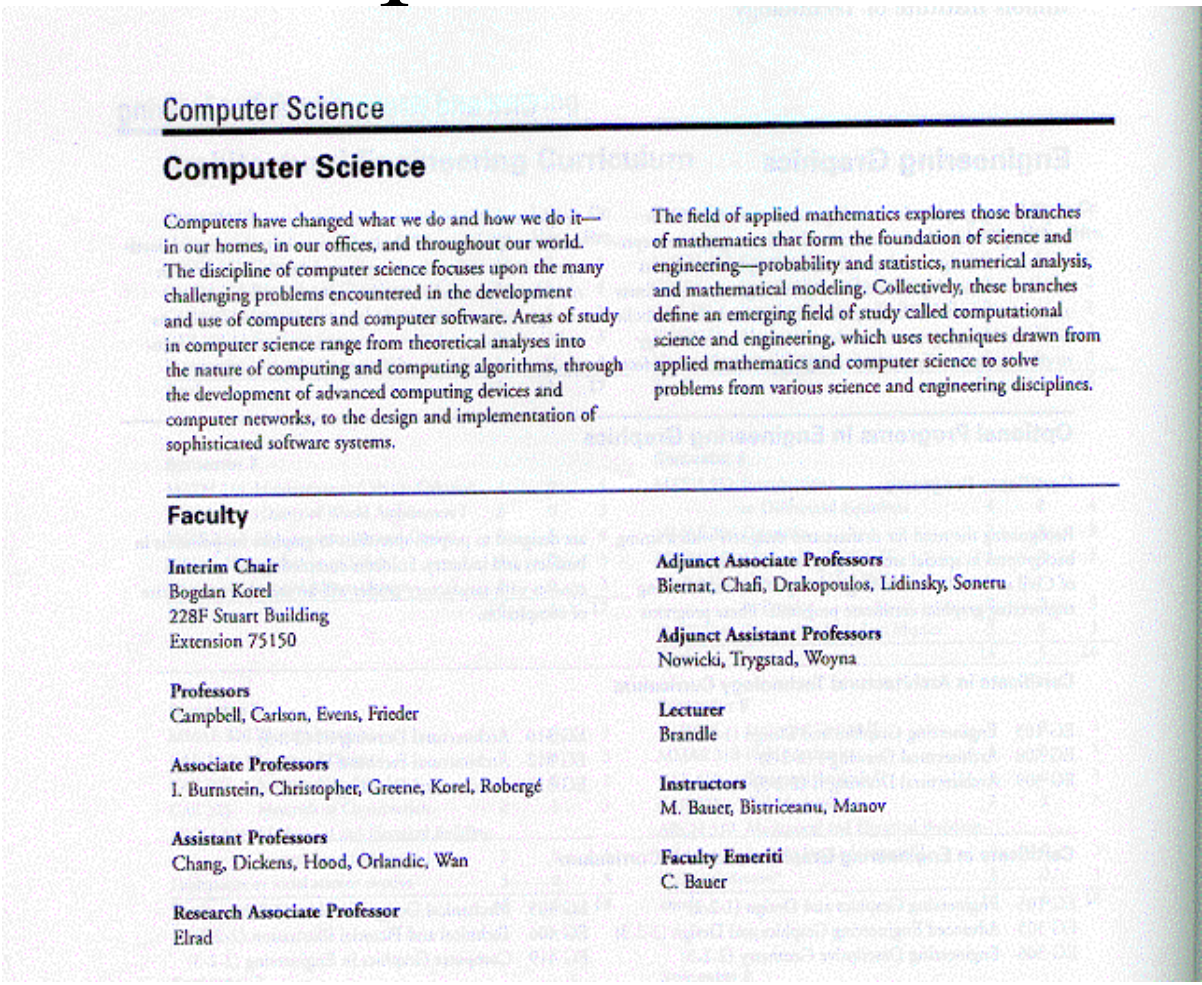
Regular Expressions

- Regular Expressions are constructs that can be used to match specific patterns within unstructured data.
- We used them to enhance the flexibility of our parsing code, making use of the 30 years of engineering that have been put into them.
- This regular expression will match the word “Semester” followed by an integer from 0-9:
  - Semester ([0-9])

Section 1: Input Data

- Name →
- Description →

- Dept. Chair →
- Faculty →



Computer Science

Computer Science

Computers have changed what we do and how we do it—in our homes, in our offices, and throughout our world. The discipline of computer science focuses upon the many challenging problems encountered in the development and use of computers and computer software. Areas of study in computer science range from theoretical analyses into the nature of computing and computing algorithms, through the development of advanced computing devices and computer networks, to the design and implementation of sophisticated software systems.

The field of applied mathematics explores those branches of mathematics that form the foundation of science and engineering—probability and statistics, numerical analysis, and mathematical modeling. Collectively, these branches define an emerging field of study called computational science and engineering, which uses techniques drawn from applied mathematics and computer science to solve problems from various science and engineering disciplines.

Faculty

Interim Chair
Bogdan Korel
228F Stuart Building
Extension 75150

Professors
Campbell, Carlson, Evens, Frieder

Associate Professors
I. Burnstein, Christopher, Greene, Korel, Robergé

Assistant Professors
Chang, Dickens, Hood, Orlandic, Wan

Research Associate Professor
Elrad

Adjunct Associate Professors
Biernat, Chafi, Drakopoulos, Lidinsky, Soneru

Adjunct Assistant Professors
Nowicki, Trygstad, Woyna

Lecturer
Brandle

Instructors
M. Bauer, Bistriceanu, Manov

Faculty Emeriti
C. Bauer

Used to Build:

- <name>

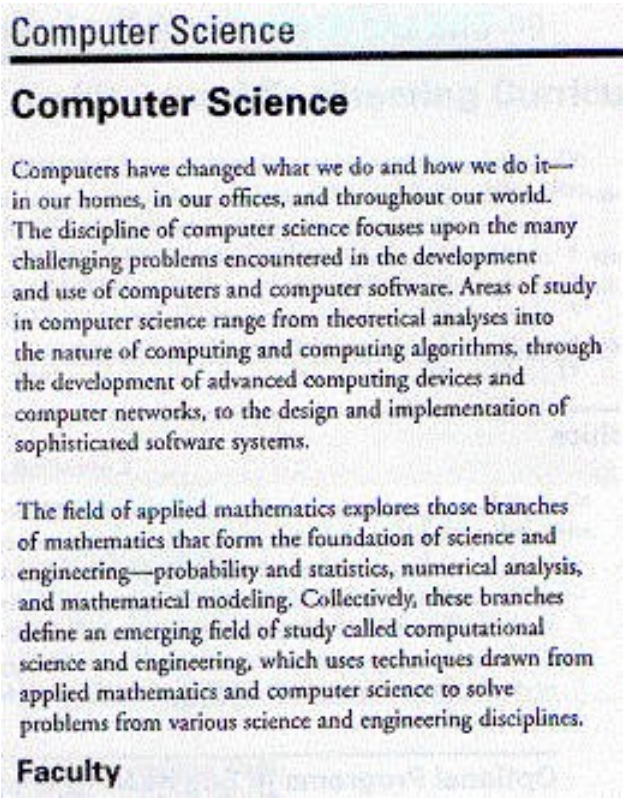
- <overview>

- <faculty>

Section 1:
Department Title and Overview

Dept. Title →

Overview →



Faculty →

(next slide)

- Straightforward problem if isolating blocks of text and processing.
- These two pieces of data had few organizational discrepancies.
- The title of each department was always on the first line of each department file.
- The overview is always located between the department title and the faculty section.
- Employed the use of Regular Expressions to selectively select parts of the whole file reliably.

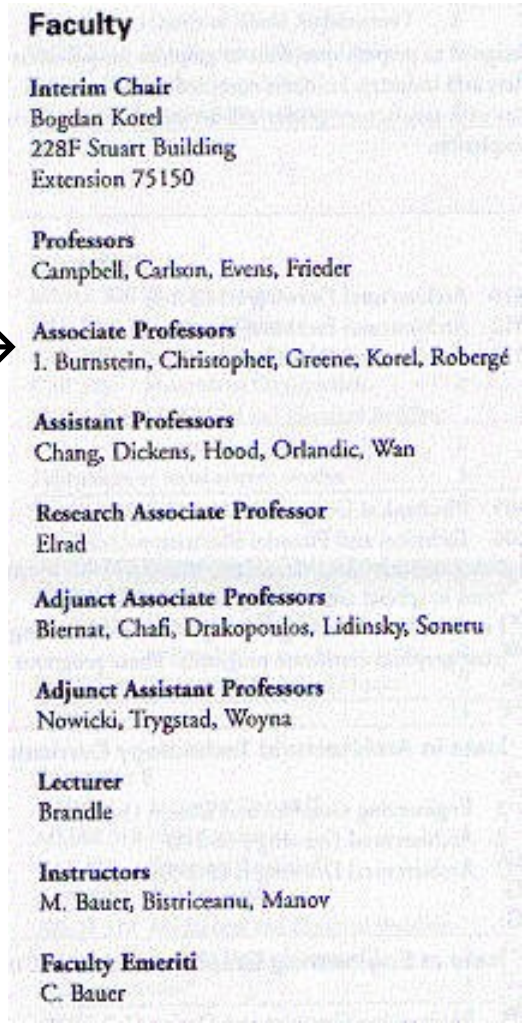
Section 1: Faculty

Faculty Start →

"Chair" entry →

Professor entries →

Faculty End →



Faculty

Interim Chair
Bogdan Korel
228F Stuart Building
Extension 75150

Professors
Campbell, Carlson, Evens, Frieder

Associate Professors
I. Burnstein, Christopher, Greene, Korel, Robergé

Assistant Professors
Chang, Dickens, Hood, Orlandic, Wan

Research Associate Professor
Elrad

Adjunct Associate Professors
Biernat, Chafi, Drakopoulos, Lidinsky, Soneru

Adjunct Assistant Professors
Nowicki, Trygstad, Woyna

Lecturer
Brandle

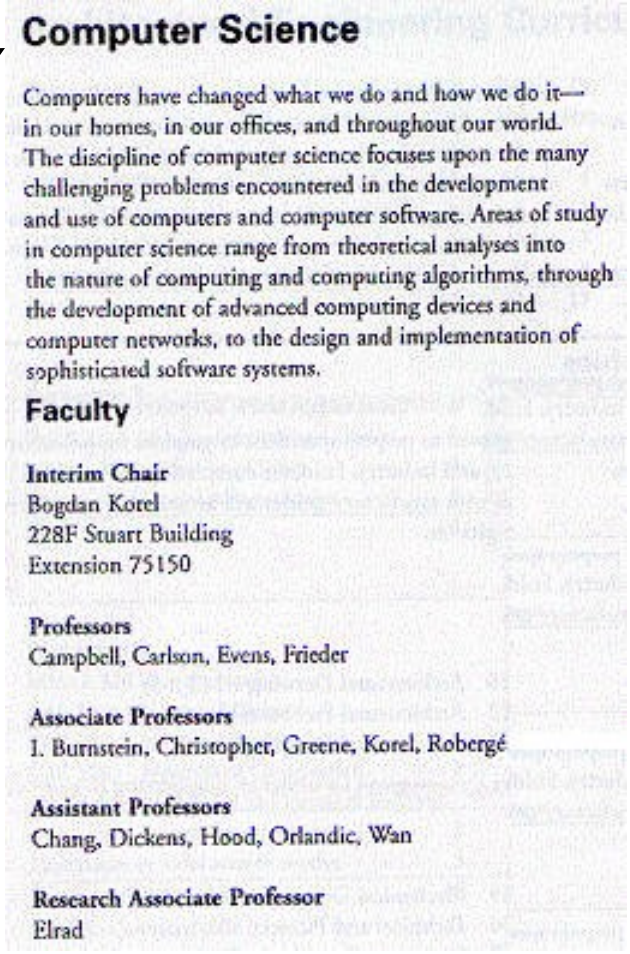
Instructors
M. Bauer, Bistriceanu, Manov

Faculty Emeriti
C. Bauer

- The Faculty section provided a more interesting challenge than the other two sections.
- Once the body of the faculty section was isolated, further sub-processing needed to be done.
- Need to detect and handle special entries (ie. Chair).
- Need to properly build all "member" sub-tags of the "faculty" tag.
- Once isolated as a whole, the entire faculty section was processed on a line-by-line basis.

Section 1: Regular Expressions

- \\A([^\n]+)\n
  - Matches First Line of Document
- \\A[^\n]+\n(.+)\n\nFaculty\n
  - Matches between first line and "Faculty"
- \\A[^\n]+\n.+\n\nFaculty\n(.+)
  - Matches after "Faculty"



Computer Science

Computers have changed what we do and how we do it—in our homes, in our offices, and throughout our world. The discipline of computer science focuses upon the many challenging problems encountered in the development and use of computers and computer software. Areas of study in computer science range from theoretical analyses into the nature of computing and computing algorithms, through the development of advanced computing devices and computer networks, to the design and implementation of sophisticated software systems.

Faculty

Interim Chair
Bogdan Korel
228F Stuart Building
Extension 75150

Professors
Campbell, Carlson, Evens, Frieder

Associate Professors
I. Burnstein, Christopher, Greene, Korel, Robergé

Assistant Professors
Chang, Dickens, Hood, Orlandic, Wan

Research Associate Professor
Elrad

Section 1:
Final Production

Computer Science

Computers have changed what we do and how we do it—in our homes, in our offices, and throughout our world. The discipline of computer science focuses upon the many challenging problems encountered in the development and use of computers and computer software. Areas of study in computer science range from theoretical analyses into the nature of computing and computing algorithms, through the development of advanced computing devices and computer networks, to the design and implementation of sophisticated software systems.

Faculty

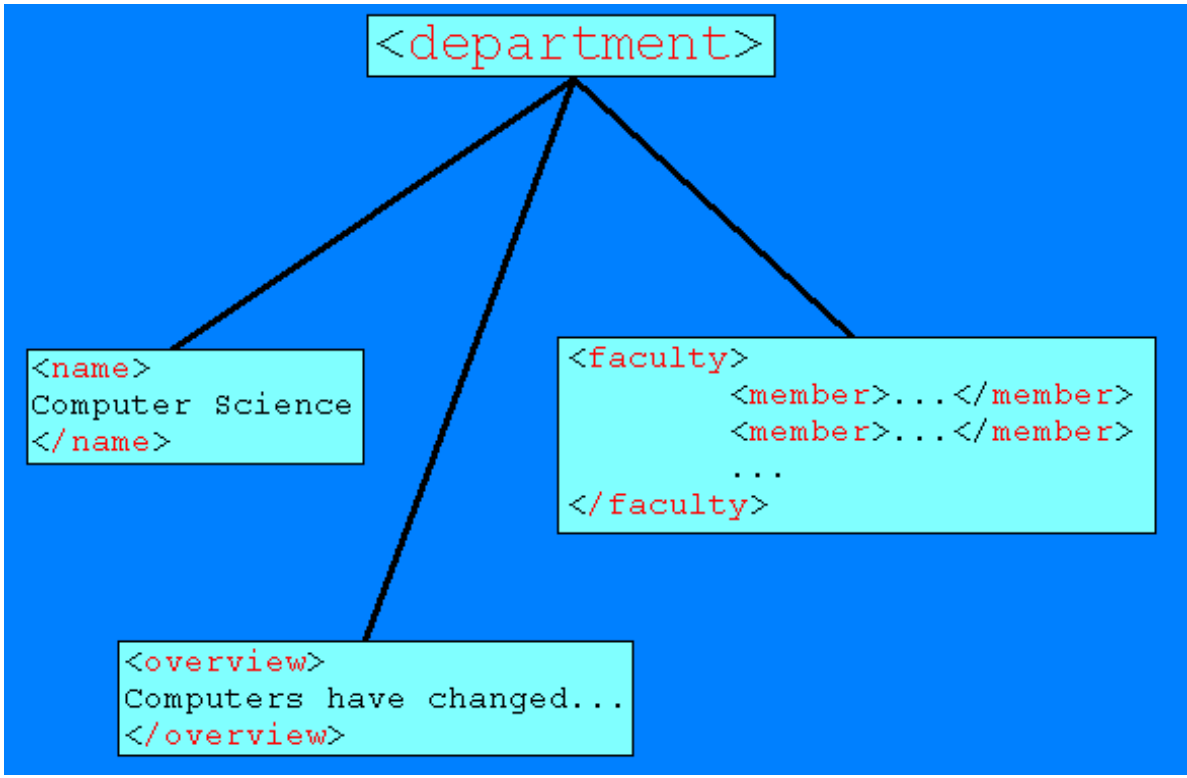
Interim Chair
Bogdan Korel
228F Stuart Building
Extension 75150

Professors
Campbell, Carlson, Evens, Frieder

Associate Professors
I. Burnstein, Christopher, Greene, Korel, Robergé

Assistant Professors
Chang, Dickens, Hood, Orlandic, Wan

Research Associate Professor
Elrad



Faculty Section: Example Output

- The faculty tag was designed to have many “member” tags
- This is a “member” tag

Raw Data:

```
Edward Reingold
236B Stuart Building
Extension 75150
```

Result:

```
<member isChair="true" rank="Chair">
        <name>Edward Reingold</name>
        <office>
                <number>236B</number>
                <building>Stuart Building</building>
        </office>
        <phone>Extension 75150</phone>
</member>
```

Section 2: Data

- Description……
- Major Title……
- Required Courses…….

Computer Science

The department offers two undergraduate programs in computer science: a Bachelor of Science in Computer Science and an Applied Science for the Professions Bachelor of Science in Computer Information Systems. Both programs provide an excellent background in computer science and allow for ample study in other areas. Where these programs differ is in the approach they take to computer science. The B.S. in Computer Science provides an in-depth experience focusing on the theory and practice of computer science while the B.S. in Computer Information Systems provides a more interdisciplinary experience, balancing study in computer science with study in another field. In addition to these programs in computer science, the Department of Computer Science and the Department of Electrical and Computer Engineering jointly offer a Bachelor of Science in Computer Engineering. This program focuses on both the digital electronics hardware used in computer systems and the software that controls this hardware, with an emphasis on the design and implementation of computer-controlled systems. This program is described in detail on page 75.

All three programs begin with a set of introductory courses that work together to provide students with a firm foundation in computer science. These introductory courses include weekly labs in which students use state-of-the-art software development techniques (object-oriented programming in C++, for instance) to create solutions to interesting problems. The department's unique four-phase laboratory model encourages student creativity by providing ample opportunity for constructive feedback on each student's efforts. Having completed the introductory core, a student is prepared to work independently within a well-structured design framework—in the classroom or on the job.

The last two years of study build upon this foundation. The Bachelor of Science in Computer Science focuses on the concepts and techniques used in the design and development of advanced software systems. Students in this program explore the conceptual underpinnings of computer science—its fundamental algorithms, programming languages, operating systems, and software engineering techniques. In addition, students choose from a rich set of electives—including computer graphics, artificial intelligence, database systems, computer architecture, and computer networks, among others. As with the introductory sequence, these advanced courses stress "hands-on" learning by doing. A generous allotment of free electives allows students to combine study in computer science with study in another field—either by taking a well-defined specialized minor in another discipline or by working with an adviser to formulate a program that combines experiences across disciplines.

The B.S. in Computer Information Systems program emphasizes the use of computers as sophisticated problem-solving tools. Students in this program pursue an interdisciplinary course of study that combines a solid foundation in computer science with a focus in another discipline. This program is designed for students who seek to blend their computer science abilities with skills specific to another domain to solve problems in that domain. Examples include computing with a business focus (e.g., management information systems) or computing with a natural science focus (e.g., computational physics).

Bachelor of Science in Computer Science

Required Courses	Credit Hours
Computer Science Requirements CS 100, 101, 105, 106, 330, 331, 350, 351, 430, 440, 450, 487	32
Computer Science Electives	15
Mathematics Requirements MATH 151, 152, 251, 474	17
Mathematics Electives	3
Science/Engineering Requirements PHYS 123, 221	8
Science/Engineering Electives	6

Required Courses	Credit Hours
Humanities Requirements PHIL 374 or CS 485	3
Humanities 100-level course	3
Humanities Electives	9
Social Science Electives (including at least three hours in economics)	12
Non-Technical Elective	3
Interprofessional Projects	6
Free Electives	12
Total Credit Hours	129

Section 2:
Major Title

- The module uses the data file which is given to me by the program which contains the major section.
- The major title is located in front of the description.
- If the description is not present in the data file then I assume that it is present before the required courses section.

Section 2: Description

- The description tag is built after the major title is built.
- This is the easiest of all the things assuming the description is available to me from the data file.
- Some files have missing descriptions. It still handles the missing sections gracefully.

Section 2:
Required Courses

- The required courses section proved very challenging.
- Dealt with requirements and electives. Which are very different.

Required Courses	Credit Hours
Computer Science Requirements CS 100, 101, 105, 106, 330, 331, 350, 351, 430, 440, 450, 487	32
Computer Science Electives	15
Mathematics Requirements MATH 151, 152, 251, 474	17
Mathematics Electives	3
Science/Engineering Requirements PHYS 123, 221	8
Science/Engineering Electives	6

Required Courses	Credit Hours
Humanities Requirements PHIL 374 or CS 485	3
Humanities 100-level course	3
Humanities Electives	9
Social Science Electives (including at least three hours in economics)	12
Non-Technical Elective	3
Interprofessional Projects	6
Free Electives	12
Total Credit Hours	129

Section 2: Example Output

- A typical sample XML file generated from the required course section.

```
<requirement discipline="Computer Science" listType="electives">
        <totalHours>15</totalHours>
    </requirement>
    <requirement discipline="Mathematics" listType="courses">
        <totalHours>17</totalHours>
        <course>
            <department>MATH</department>
            <number>151</number>
        </course>
        <course>
            <department>MATH</department>
            <number>152</number>
        </course>
        <course>
            <department>MATH</department>
            <number>251</number>
        </course>
        <course>
            <department>MATH</department>
            <number>474</number>
        </course>
    </requirement>
```

Input Data

Computer Science Electives	15
Mathematics Requirements	17
MATH 151, 152, 251, 474	

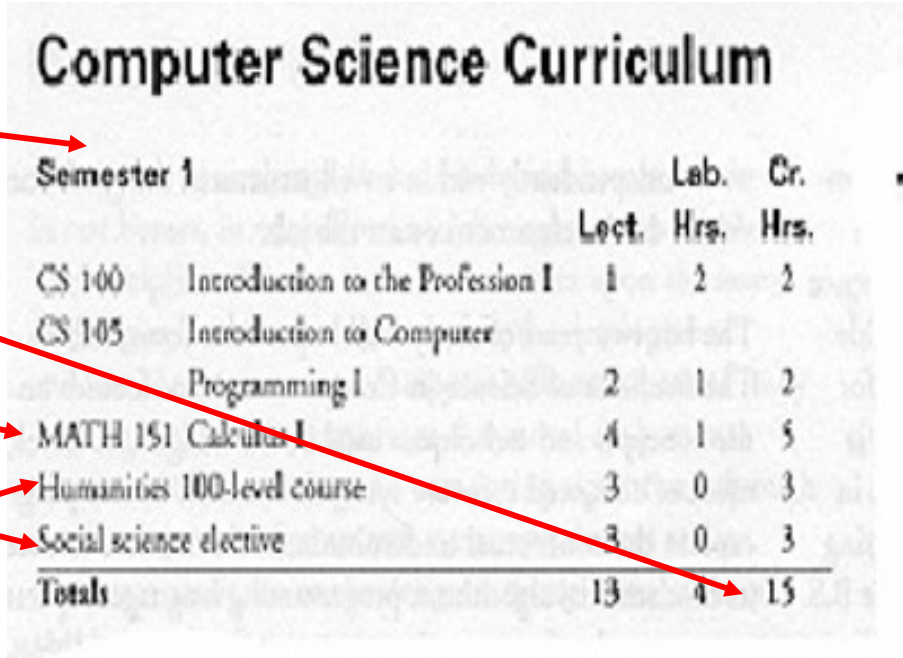
Section 2: Discrepancies

- Assumed all the data files are in the format of Computer Science curriculum.
- eg. Of other data files which had discrepancies.

- Psychology Requirements 33
- PSYC 204, 221, 222, 301, 303, 406, 435 |or 436, 482, 483, 487, 488
- Introduction to the Profession 100 (2 semesters) 4
- Psychology Electives 15
- Mathematics Requirements 6
- MATH 122, 221
- Computer Science Requirement 2
- CS 105
- Natural Sciences Requirements 12-13
- CHEM 124, BIOL 107 and/or 115*, PHYS 211

Section 3:
Sample Curriculum

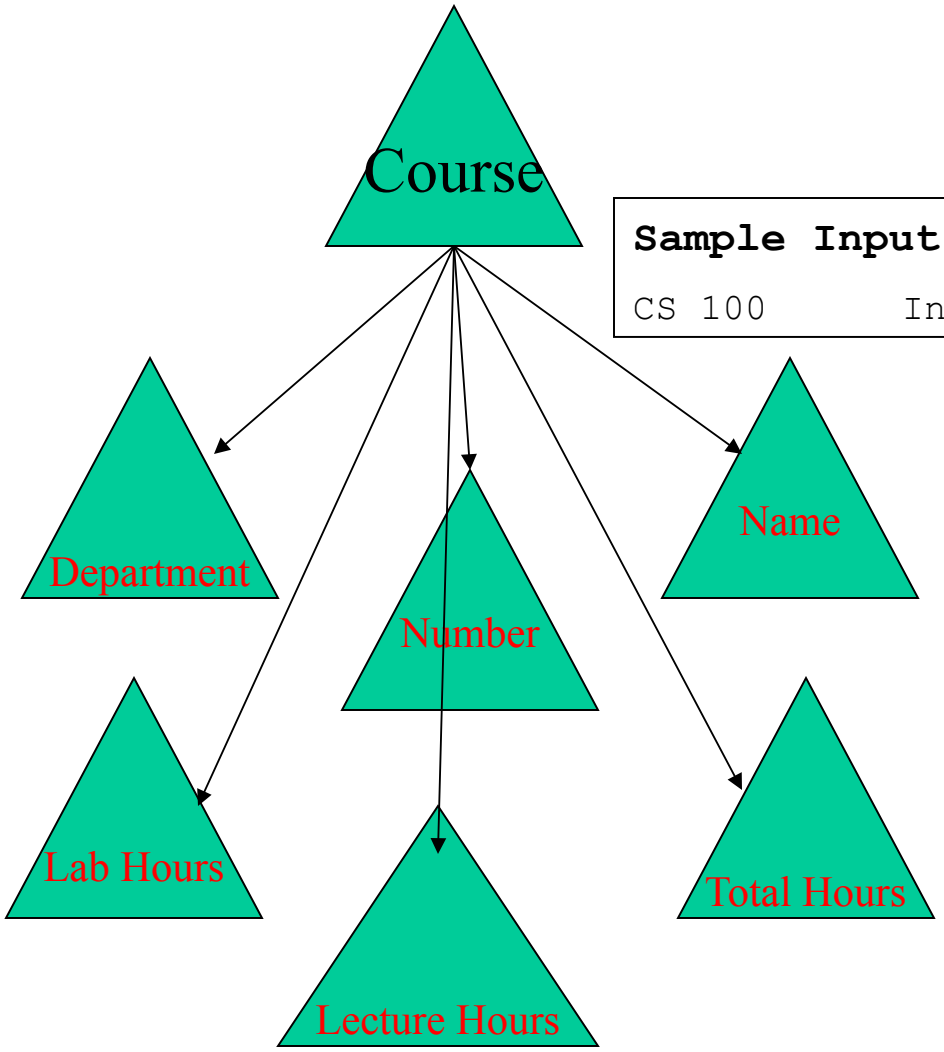
- Semester
  - Semester number
  - Total hours
  - Courses
  - Electives
  - Special Electives



Computer Science Curriculum

Semester 1		Lect.	Lab. Hrs.	Cr. Hrs.
CS 100	Introduction to the Profession I	1	2	2
CS 105	Introduction to Computer Programming I	2	1	2
MATH 151	Calculus I	4	1	5
Humanities	100-level course	3	0	3
Social science	elective	3	0	3
Totals		13	4	15

Section 3: Courses



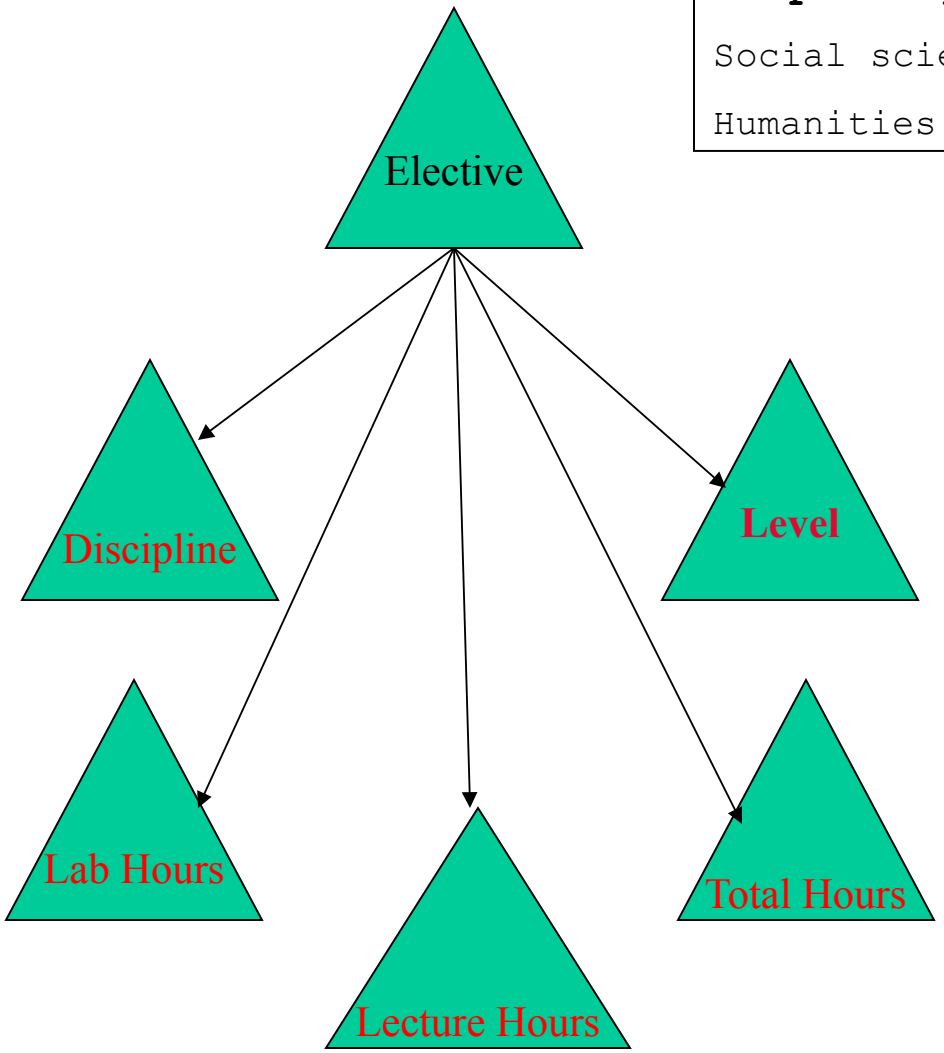
Sample Input data

```
CS 100      Introduction to the Profession I  1   2   2
```

Sample Output

```
<course>
    <department>CS</department>
    <number>100</number>
    <name>Introduction to the Profession I</name>
    <labHours>2</labHours>
    <lectureHours>1</lectureHours>
    <totalHours>2</totalHours>
</course>
```

Section 3: Electives



Sample Input Data

```
Social science elective        3    0     3
Humanities 100-level course    3    0     3
```

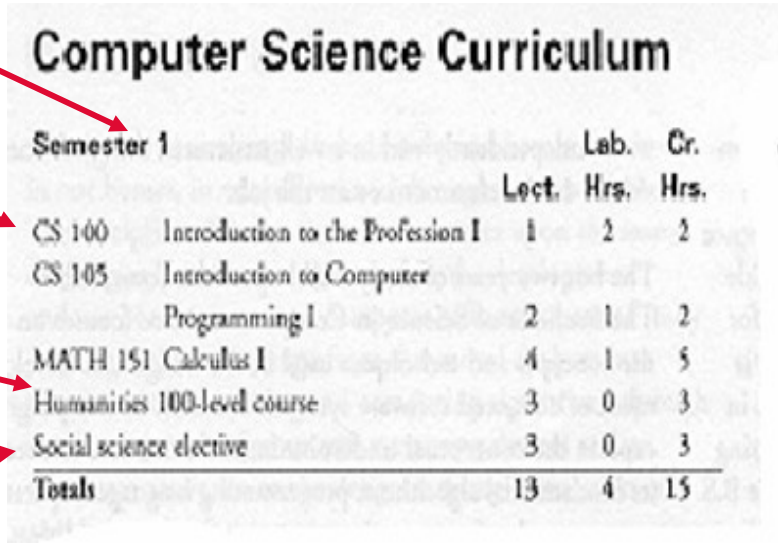
Sample Output

```
<elective>
<discipline>Social science</discipline>
<labHours>0</labHours>
<lectureHours>3</lectureHours>
<totalHours>3</totalHours>
</elective>
```

```
<elective>
<discipline>Humanities</discipline>
<level>100</level>
<labHours>0</labHours>
<lectureHours>3</lectureHours>
<totalHours>3</totalHours>
</elective>
```

Section 3: Example Output

```
<sampleCurriculum>
    <semester number="1">
            <course>
                    <department>CS</department>
                    <number>100</number>
                    <name>Introduction to the Profession I</name>
                    <labHours>2</labHours>
                    <lectureHours>1</lectureHours>
                    <totalHours>2</totalHours>
            </course>
.....
             <elective>
                    <discipline>Humanities</discipline>
                    <level>100</level>
                    <labHours>0</labHours>
                    <lectureHours>3</lectureHours>
                    <totalHours>3</totalHours>
            </elective>
            .....
            <elective>
                    <discipline>Social science</discipline>
                    <labHours>0</labHours>
                    <lectureHours>3</lectureHours>
                    <totalHours>3</totalHours>
            </elective>
            <totalHours>15</totalHours>
    </semester>
</sampleCurriculum>
```

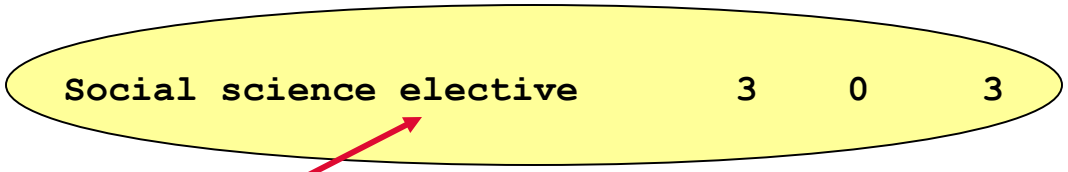


Computer Science Curriculum

Semester 1		Lect.	Lab. Hrs.	Cr. Hrs.
CS 100	Introduction to the Profession I	1	2	2
CS 105	Introduction to Computer Programming I	2	1	2
MATH 151	Calculus I	4	1	5
Humanities 100-level course		3	0	3
Social science elective		3	0	3
Totals		13	4	15

Section 3: Regular Expressions

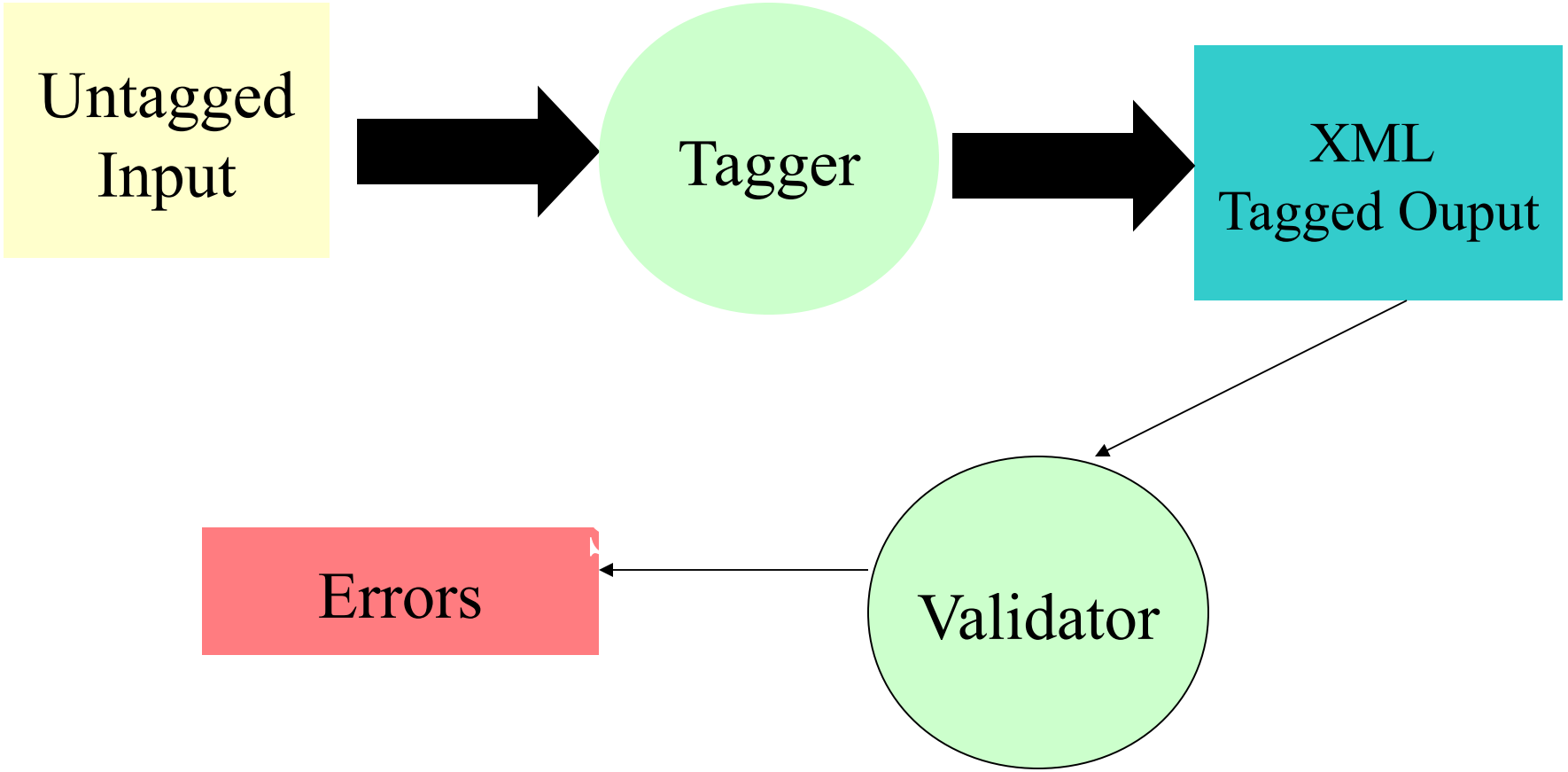
- Some regular expressions for this section:
  - **Semester Number**
    - ^Semester ([0-9]+)
  - **Course String**
    - ^([A-Z]+) +([0-9]{3}) +([A-Z].*[A-Za-z]) +([0-9]*) +([0-9]*) +([0-9]*)$
  - **IPRO String**
    - ^([A-Z]+) +([I]{1,2}) +([A-Z].*[A-Za-z]) +([0-9]*) +([0-9]*) +([0-9]*)$
  - **Elective Course String**
    - ^(.+ [Ee]lective[s]*) +([0-9]*) +([0-9]*) +([0-9]*)$
  - **Special Elective Course String**
    - ^(.+) +([1-7]00)-level course +([0-9]*) +([0-9]*) +([0-9]*)$
  - **Total Semester Hours**
    - ^Totals +([0-9]{1,2}*) +([0-9]{1,2}*) +([0-9]{1,2}*)$



Testing Phase

- Check to ensure continuity between the source data and the produced XML.
- Check text format to update schema of new data.
- Check to ensure that the Tagger produces valid XML files.
- Report any errors identified for correction.
- For xx files, we have yy missing fields, zz errors, etc.

Testing Process



User Interface Testing

- Planned user interface testing with IPRO xxxx.
- Some meetings with IPRO xxxx provided helpful requirements to facilitate testing.
- Requirements were met, but too late for any user interface testing this semester.
- Input from user interface team has already helped our prototype.

Summary & Future Work

- We developed a functional prototype for tagging the IIT Undergraduate Bulletin with XML.
- The produced XML files can be used as a data source for our mediator.
- Our mediator now has support for semi-structured queries, adding yet another dimension to its search capabilities.
- In the Spring we hope to extend the tagger so that more university data can be searched with our mediator.