CIS 895 – MSE Project

KDD-Research Entity Search Tool (KREST)
Presentation 2

Eric Davis
efd3467@ksu.edu
Outline

- Action Items
- Architectural Design
- Test Plan
- Formal Inspection Checklist
- Project Plan
- Prototype Demonstration
- Questions / Comments
Create a substantial formal specification for the project

- Entire project has been formally specified
  - Involves gets/sets as well as additions, deletions, and searching from the database.
  - Formal Specified in OCL by hand and checked using USE 2.3.1.
- Emailed committee to notify of the formal specification status on 12/07/07.
Action Items (cont.)

- Investigate Effort Adjustment Factors (EAFs) for Complexity and Data.
  - Complexity and Data EAFs remain at High
    - The largest possible value for the two factors.
    - Actual storage requirements are linear, based on the number of websites that the user wishes to crawl.
      - Would be exponential if unrestricted.
Action Items (cont.)

- Investigate depth-limited crawlers (Wget, Teleport Crawl, etc).
  - COTS crawlers provide crawling ability, ability to limit by depth, and follow Robot Exclusionary Protocol.
  - Decision made to implement crawler, rather than use COTS
    - Allows developer to learn about web crawling
    - Majority of crawler code already developed
      - Depth limited crawling was added for second demo
    - Use of a COTS crawler may be a nice add-on for future work
Move ‘Minimum # of back links’ field to from the Crawler tab to the Web Search tab.

- The back links field was moved to the Web Search tab.
Action Items (cont.)

- Delineate scope of KREST in comparison to Tao Cheng’s Entity Search work.
  - Differences of KREST:
    - GUI based
    - Able to run on a single PC / Linux machine
      - No need for a cluster
    - Will be run on smaller datasets
    - Limited to contact information entities
    - No complex algorithm for ranking entities found
A Model-View-Controller (MVC) approach was used

Developed using MS Visio

Class Descriptions, Attributes and Operations are contained in Architecture Design Document
Architectural Design

- Overall Package View
Architectural Design

- Controller Package:
Architectural Design

- View Package:
Architectural Design

- Model Package:
Sequence Diagram – Performing a Web Crawl:
Architectural Design

- Sequence Diagram – Performing a Web Search:
Architectural Design

- Sequence Diagram – Performing an Entity Search:
Architectural Design

- Formal Specification
  - Created and validated using USE 2.3.1.
  - All Classes are specified
    - All important attributes and methods are specified
    - Get() methods of Java specific GUI features are not specified
  - Contained at the end of the Architectural Design Document
  - 14 associations, 22 invariants, 87 pre/post conditions
Test Plan

- Functional, black-box testing will be performed
- Testing broken into five test cases:
  - Application Requirements
  - Web Crawl Requirements
  - Web Search Requirements
  - Entity Search Requirements
  - Reproducing similar results to Tao Cheng’s work
- Each step in the test cases include:
  - Tester actions
  - Expected results
  - Requirement numbers mapped to the expected results
- Test Plan also lists dependencies between the test cases for Formal Testing
The following items are to be checked:

- The symbols used in the class diagrams conform to UML standards.
- The symbols used in the sequence diagrams conform to UML standards.
- The classes in the class diagrams have corresponding descriptions provided in the Architecture Document.
- The descriptions of the classes in the Architecture Document are clear and concise.
- The classes in the USE model are consistent with those in the Architecture Document.
The following items are to be checked:

- The attributes in the USE model are consistent with the attributes of the corresponding class diagrams.
- The associations in the USE model are present in the class diagrams as association links.
- The multiplicities in the USE model are consistent with the multiplicities of the corresponding class diagrams.
- The sequence diagrams are clear and concise.
- All model elements outlined in the Vision Document are present in the Architecture Document as classes.
Goal: To be completely done with all docs submitted by May 2, 2008
Project Schedule (cont.)
Project Plan

- **Current Status**
  - 2K SLOC developed
  - 29/34 Requirements Implemented = 85 %
  - Productivity = 17.86 LOC/HR
    - \[ \frac{2000}{\left(\frac{6720}{60}\right)} = 17.86 \]
  - Code Remaining = 353 LOC
    - \[ \frac{2000}{0.85} - 2000 = 353 \]
  - Time Remaining = 20 Hours
    - \[ \frac{353}{17.86} = 20 \]
Project Plan (cont.)

- Remaining Effort
  - Coding: 20 / 2 Hr/Day = 10 Days
  - Testing: 21 Days
  - Documentation: 25 Days

- Total of 56 days – would place completion at April 9th
  - 16 days ahead of original estimate
Prototype Demonstration
Phase 3 Deliverables

- Action Items
- Component Design
- Assessment Evaluation
- Project Evaluation
- User’s Manual
- Formal Technical Inspection Letters
- Presentation 3
- Source Code + JavaDoc
- Executable Project
- Portfolio
Current Obstacles / Questions

- Technical Inspectors
  - One is still needed

- Presentation 3 Date
  - Goal: Have ‘Approval to schedule final exam’ form submitted by Apr. 4th for inclusion in commencement documents
    - Draft Portfolio to committee by March 30th
    - Presentation by Wednesday, April 23rd
    - Final portfolio submitted by May 2nd

- ‘Final Exam Form’
  - Requires courses from previous semesters have grades (i.e. no incompletes)
  - Will Fall Semester CIS 895 be an issue?
Questions / Comments