Multi-agent Research Tool (MART)
A proposal for MSE project

Madhukar Kumar
Agenda

- Proposal – Project statement and overview
- Requirements – Use case, requirement specifications
- Cost Estimation
- Project Plan
- Software Quality Assurance Plan
- Architecture Elaboration Plan
- Demonstration
Project Statement

- Multi-agent system that helps a researcher/author in writing an article by automatically searching and dynamically generating links while an article is being written.
- Intended audience
  - Reporters/content managers
  - Researchers
- The system will be built using the MaSE methodology and agentTool.
- Why Multi-agent and why MaSE?
Project Overview
Requirements – Use Case

System (MS Word and MART)

- Type Article
- Search
- Display Results

Researcher
Use Case 1: Type article in MS Word

- Description - The user should be able to type an article in MS Word with his/her normal settings and preferences.

- Specific requirements -
  - Features - When MART is finally integrated with MS Word, none of the features, preferences and settings should either change or stop working.
  - Save - The user should also be able to save the articles as any other regular files.
Use Case 2: Search for research material

- Description – The user should be able to click on a button somewhere in the MS Word menu that should trigger the MART search agents.

- Specific requirements –
  - Filter – Once the search button is pressed, all the irrelevant words like “a”, “the,” “an”, etc should be removed and a list of keywords should be prepared for searching database.
  - Relevance – The list of keywords should be ranked according to the number of times each word has been used in the typed article.
Critical Use Cases (3)

- **Search** – Using the list of relevant keywords, one Agent should crawl on an online search engine, conduct a search and return with html content that should be saved as a file named as results.html in My Documents folder of the user’s computer.

- **Customize display** – The content in results.html should be re-arranged by inserting appropriate html tags so that only the first 10 relevant links are shown in the web page. Clicking on these links should take the user directly to the information source.
Use Case 3: Display research material

- Description – The user should be able to click on a button somewhere in the MS Word menu that should immediately open a window of Internet Explorer ® browser that will have all links to web sites with information relevant to the current article being typed in MS Word.

- Specific requirements –

  - Display – The results.html file should be opened in Internet Explorer browser.
## Cost Estimation (using Function Point)

<table>
<thead>
<tr>
<th>Type</th>
<th>Simple</th>
<th>Average</th>
<th>Complex</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td></td>
<td></td>
<td>1. Entire Word file being typed by user</td>
<td>6</td>
</tr>
<tr>
<td>Inquiries</td>
<td></td>
<td></td>
<td>1. Search Keywords</td>
<td>6</td>
</tr>
<tr>
<td>Files</td>
<td></td>
<td>1. Results file (HTML) 2. Text file of relevant keywords</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Interfaces</td>
<td>1. Microsoft Word 2. Internet Explorer</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>55</td>
</tr>
</tbody>
</table>
Cost Estimation (using COCOMO)

- **Organic Mode**

- **Effort (in units of programmer months)** = 2.4 * (KDSI) \(^{1.05}\)
  - KDSI = 1.5 K (conversion from function point of 55)
  - Also consistent with past experience in using agentTool – nine classes X 160 lines of code.
  - Effort or PM = 3.67

- **TDEV (Development time)** = 2.5* (PM) \(^{0.38}\)
  - Duration or TDEV = 4.09 months.
◆ Inception phase
  ■ Develop overall requirements
  ■ Develop prototype
  ■ Artifacts – vision document, project plan and software quality assurance plan.
  ■ **Milestone – Presentation 1** – Get approval from committee and incorporate changes and suggestions.

◆ Elaboration Phase
  ■ Develop Goals
  ■ Develop Roles
  ■ Develop Concurrent tasks
  ■ Develop Agent Template
  ■ Develop Conversations
  ■ Verify Conversations
  ■ Develop Deployment scheme
  ■ Refine vision document, project plan and software quality assurance plan.
  ■ **Milestone – Presentation 2** – Get approval from committee and/or incorporate changes in the design.
**Production Phase**
- Develop User Interface
- Generate code stub
- Integrate

- **Testing**
  - Functional testing

- **Documentation**
  - Java source code with Javadoc API HTML documents.
  - Design artifacts – object model etc.
  - Test report.
  - Evaluation report.

- **Milestone – Presentation 3** -- Get approval from committee and/or incorporate changes/suggestions.
SQA (Resources)

- **Reference Material**

- **Software Development Environment**
  - agentTool 1.8.3
  - Java 1.4.1 (including java.net packages)
  - Visual Basic Application (Visual Studio .Net 7.0.9466)
  - MS Visio 10.0.525 (for developing object model and other design artifacts)
  - USE 2.0.1
SQA (Deliverables)

- agentTool maml file.
- Java source code with Javadoc API HTML documents.
- Source code for MS Word macros.
- Design artifacts – object model, use case documents.
- Vision document.
- Project Plan document.
- Software Quality Assurance (SQA) plan document.
- Test report.
The object diagram of MART generated during the elaboration phase will be formalized using the Object Constraint Language (OCL). This will be done using the USE tool.

Development of a second prototype

Development of the second prototype should establish the fact that agents can talk to each other and return with some relevant search details if some keywords are passed as search parameters.

Test plan

Formal Technical Inspection – The object diagram shall undergo formal technical inspection by two MSE students who have agreed to participate in the inspection - Cem Oguzhan and Esteban Guillen.
Constraints

- Since the Agent Tool generates stub code in Java, the program will be developed in Java.
- The Google API only returns the first 10 search results per query.
- There is a limit to the number of searches a program can conduct every day using the Google API.
Prototype Demonstration

- Demonstration of prototype