# Agent-based Metamodelling

**Sandhya Madireddy**

## SCHEDULE

<table>
<thead>
<tr>
<th>Key development milestones for 2004:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop rules for the meta model</td>
</tr>
<tr>
<td>Proposal Presentation</td>
</tr>
<tr>
<td>Reformat the data</td>
</tr>
<tr>
<td>Develop agent to retrieve the distributed data</td>
</tr>
<tr>
<td>Compare the model runs</td>
</tr>
<tr>
<td>Develop learning agents for the sensitivity analysis</td>
</tr>
<tr>
<td>Develop the user interface</td>
</tr>
<tr>
<td>Case Study with Demo</td>
</tr>
<tr>
<td>Thesis Document</td>
</tr>
<tr>
<td>Thesis Defense</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>10, June 2004</td>
</tr>
<tr>
<td>15, June 2004</td>
</tr>
<tr>
<td>20, June 2004</td>
</tr>
<tr>
<td>05, July 2004</td>
</tr>
<tr>
<td>20, July 2004</td>
</tr>
<tr>
<td>20, August 2004</td>
</tr>
<tr>
<td>01, Sept 2004</td>
</tr>
<tr>
<td>10, Sept 2004</td>
</tr>
<tr>
<td>30, Oct 2004</td>
</tr>
<tr>
<td>Nov 1st week</td>
</tr>
</tbody>
</table>

## OBJECTIVE/APPROACH

**Objective:**
- Make the model flexible: run with distributed heterogeneous data
- Transform the data into the required formats
- History analysis: compare all the runs of the model at different times and provide a feedback for the user
- Sensitivity analysis: advise what the input should be to get the desired output

**Approach:**
- Create framework-based agents:
  - to retrieve the distributed data
  - to compare the different runs of the model, and
  - to analyze input and output data for history and sensitivity analysis
- Develop rule based meta model
- Develop a meta model driven friendly user interface

## MISCELLANEOUS SUPPORTING DATA

**Benefits:**
1. Enhancing a family of agent-based models by providing meta modeling framework.
2. Rules in the meta model are independent of the platform and the software or hardware used.
3. Feeding data into the model from multiple distributes sources.
4. Enhancing the custom model by providing pluggable analysis logic.
5. Enhancing the custom model by providing pluggable preprocessing logic.
6. Friendly user interfaces with generic meta model-driven features.

---

**Diagram:**

```
  Data -> Pre/Post-processing Agent -> Dispatching Agent -> Meta-model
          |                                    |                        |
          |                                    |                        |
Custom Plug-ins  |  |  |  | Analysis Agent
          |                                    |                        |
          |                                    |                        |
Hurley Pasture Model
```

---

**THESIS DESCRIPTION**

In the modeling systems where data is distributed and multiple components collaborate, a generic framework is necessary to simplify deployment of new models. The framework is rule-based and also includes user interfaces. The Hurley Pasture model is used as a case study.

---

**Data Pre/Post-processing Agent**

```
  Data
  |-> Pre/Post-processing Agent
```

---

**User Interface**

```
  User Interface
  |-> Analysis Agent
```

---

**Hurley Pasture Model**

```
  Hurley Pasture Model
  |-> Meta-model
```

---

**Custom Plug-ins**

```
  Custom Plug-ins
  |-> Meta-model
```

---

**Dispatching Agent**

```
  Dispatching Agent
  |-> Meta-model
```

---

**Analysis Agent**

```
  Analysis Agent
  |-> Meta-model
```

---

**Custom Plug-ins**

```
  Custom Plug-ins
  |-> Meta-model
```

---

**Objectives:**

- Make the model flexible: run with distributed heterogeneous data
- Transform the data into the required formats
- History analysis: compare all the runs of the model at different times and provide a feedback for the user
- Sensitivity analysis: advise what the input should be to get the desired output

---

**Approach:**

- Create framework-based agents:
  - to retrieve the distributed data
  - to compare the different runs of the model, and
  - to analyze input and output data for history and sensitivity analysis
- Develop rule based meta model
- Develop a meta model driven friendly user interface