The Class Point Approach was first started with the aim of size estimation of object-oriented products based on design documentation. Class Point Approach (CPA) quantifies classes which are the logical building blocks in object oriented paradigm as opposed to functions in procedural paradigm for which Function Point Analysis (FPA) is used. However, like FPA, CPA also depends on classification of complexity of class based on discrete intervals and hence suffers from the shortcomings of conventional FPA namely inability to capture imprecision of human knowledge. Thus our approach is based on Fuzzy Logic that allows for a gradation of values and is more tolerant to uncertainty, imprecision, partial truth and approximation to achieve tractability, robustness and low cost solution.

### Why Fuzzy Class Point?
- Captures human way of thinking
- With traditional approach, values like 20, 21 which are intuitively similar may get categorized in different classes.
- Provides a more uniform criterion for determining whether a system has very high/high/low complexity.
- Based on membership value in each complexity level, class points for a class are calculated.

#### Example:

<table>
<thead>
<tr>
<th>Class</th>
<th>Type</th>
<th>NEM</th>
<th>NSR</th>
<th>Complexity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>PDT</td>
<td>1</td>
<td>1</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>PDT</td>
<td>1</td>
<td>1</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>PDT</td>
<td>9</td>
<td>1</td>
<td>Average</td>
<td>6</td>
</tr>
</tbody>
</table>

- Different weights assigned to class B & C even though difference in NEMs is less.
- Same weights assigned to A & B even though difference in NEMs is more.

### Methodology

1. Process the information available for size estimation as follows:
   a. Identify and organize the classes into groups depending upon their application domain.
   b. Evaluate the Class Point of each class using NEM,NSR. (and NOA also for CP2).

   e.g. For class of type PDT with \( \mu_{ND} = 0.3 \) & \( \mu_{NSR} = 0.7 \)

   \[
   \text{Class Points} = 0.3 \times 6 + 0.7 \times 10 = 8.8
   \]

2. Evaluate the final Class Point value based on the results of the above steps.
   \[\text{CP} = \text{TUCP} \times \text{TCF}\]

3. Estimate the Total Unadjusted Class Point (TUCP) as the sum of class points of each class.

### Terminology

#### Types of Classes:
- PDT: Problem Domain Type
- HIT: Human Interaction Type
- DMT: Data Management Type
- TMT: Task Management Type

#### Variables:
- NEM: Number of external methods in a class
- NSR: Number of services requested
- NOA: Number of attributes in a class

#### Class Point Measures:
- CP1: Used as initial estimate. Considers NSR & NEM for computing complexity of a class
- CP2: Considers NOA, NEM, NSR.

### Conclusions and Future work

- It is proposed that use of Fuzzy Class Point Approach would yield better results than traditional methods.
- The computations for same have been implemented and membership functions generated in MATLAB.
- This can be further analyzed and proved if real data for Class Point Approach is available.
- Once the Class Points have been calculated, they can be used to compute development effort.
- One example of such work is that of Kammani et al. which compares the Fuzzy Subtractive Clustering and Artificial Neural Networks to estimate the development effort of OO systems using Class Points. However, effort can also be computed by other methods also using Class Points.

#### References