# Review Score Aggregator Requirements Specification

## 1. Scope

### 1.1 Overview

During a contest a submitter's work will be reviewed by multiple reviewers. The component provides the mechanism to aggregate the scores from individual reviewers and determine the placements. Extra logic will be involved if the aggregated scores result in a tie. The component also includes default implementations for score aggregation and tie breaker.

### 1.2 Logic Requirements

#### *1.2.1 Score Aggregation*

Multiple reviewers will provide scores for a single submitter. A final score will be calculated based on the scores. The mechanism to aggregate the final score should be pluggable.

A default implementation should be provided, which uses the arithmetic average of the scores from each reviewer.

#### *1.2.2 Placement Assignment*

Submitters will be assigned with placements starting from 1. The placements are determined by their final scores.

#### *1.2.3 Tie Breaker*

If two or more submitters come with the same final score, tie breaker mechanisms will be applied. The tie breaker mechanism can either break the ties or make the submitters share the same placement. Tie breaker mechanism should be pluggable.

A default implementation should be provided, which will compare the scores from each individual reviewer. The submitter with more wins from the individual reviewers will be assigned with the better placement. If the tie still remains, the same placements are assigned.

### 1.3 Required Algorithms

No specific algorithms are required.

### 1.4 Example of the Software Usage

An application can use the component to aggregate scores from individual reviewers and display the overall score and placement to the audiences.

### 1.5 Future Component Direction

More aggregation and tie breaker mechanisms can be provided per scenario. For instance the aggregation can be performed with the highest and lowest scores removed first.

## 2. Interface Requirements

#### *2.1.1 Graphical User Interface Requirements*

None.

#### *2.1.2 External Interfaces*

None.

#### *2.1.3 Environment Requirements*

- Development language: Java1.4
- Compile target: Java1.4

*2.1.4 Package Structure*

com.topcoder.management.review.scoreaggregator

# 3. Software Requirements

## 3.1 Administration Requirements

*3.1.1 What elements of the application need to be configurable?*

- Score aggregation mechanism
- Tie breaker mechanism

## 3.2 Technical Constraints

*3.2.1 Are there particular frameworks or standards that are required?*

None.

*3.2.2 TopCoder Software Component Dependencies:*

- Configuration Manager

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

*3.2.3 Third Party Component, Library, or Product Dependencies:*

None.

*3.2.4 QA Environment:*

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

## 3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines.

## 3.4 Required Documentation

*3.4.1 Design Documentation*

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

*3.4.2 Help / User Documentation*

- Design documents must clearly define intended component usage in the ‘Documentation’ tab of Poseidon.