### CRC-R Template

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Give the component an easy-to-remember name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>List the responsibilities assigned to the component</td>
</tr>
<tr>
<td>Collaborators</td>
<td>List of other components this component depends on for (delegated) services [out-ports]</td>
</tr>
<tr>
<td>Rationale</td>
<td>State the rationale for allocating responsibilities to this component. Provide traceability to functional requirements and qualities or meta-architecture.</td>
</tr>
<tr>
<td>Issues and Notes</td>
<td>List assumptions, constraints, unknowns, etc.</td>
</tr>
</tbody>
</table>
CRC-R Template

Purpose of the CRC-R Template

- Informally document the component early in the structuring phase.
- Provide a summary view of the component specification.

Guidelines for creating CRC-Rs

For each component

- Give the component a memorable name
- Make a first pass at assigning responsibilities to the component, taking into account architecting principles such as loose coupling and high cohesion. (Note: The responsibilities will be refined during the logical architecture phase, using component collaboration diagrams to explore the assignment of responsibilities. Remember to update the CRC-R.)
- Make a first pass at determining which components this component will collaborate with as it fulfills its assigned responsibilities. (Update the CRC-R during logical architecture.)
- Record your rationale for assigning the responsibilities to the component, so that you won’t forget, and others will understand your reasoning.
- Identify issues, assumptions, etc. relating to the component.

Uses of an CRC-R Component Description

To the architecture team

- forms a starting point for logical architecture

To others

- forms an easy-to-grasp description of the component and its responsibilities