7.3 Comparator-Checked Fault-Tolerant System

Intent

Structure a system so that an independent failure of one component will be detected quickly and so that an independent single-component failure will not cause a system failure.

Also Known As

Tandem system

Motivation

It is sometimes very important to detect component faults quickly, or to detect component faults at a specific point during processing, to prevent component faults from causing system failures. Inspection of the output of a component may not directly reveal whether a fault has occurred or not. Some mechanism is required to support detection of faults which have not yet caused a failure.

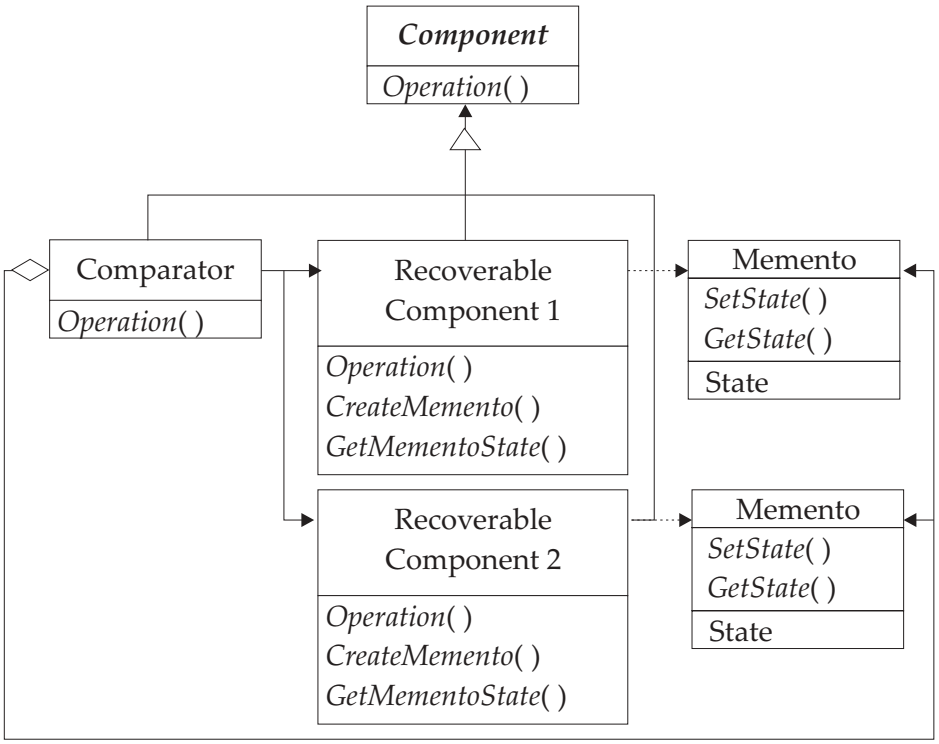
Applicability

Use Comparator-Checked Fault-Tolerant System when:

- Faults in one component are not expected to be strongly correlated with similar or identical faults in another component (this will usually be the case when faults are caused by factors external to components; it will often not be the case when faults are caused by component design or implementation errors).
- It is feasible to compare the outputs or internal states of components.
- Component faults must be detected soon after they occur, or at specific points during processing, but in any case before they lead to a system failure.
- Duplicating system components is economical.

Structure

A Comparator-Checked Fault-Tolerant System consists of an even number of Recoverable Components [TG_SDP] (often four or more), organized as sets of pairs, together with a Comparator for each pair. Each comparator examines Mementos [GoF] produced by each member of its pair to determine whether they match. If the Mementos do not match, the Comparator concludes that a fault has occurred in one of the components and takes corrective action.



Participants

- Recoverable Components

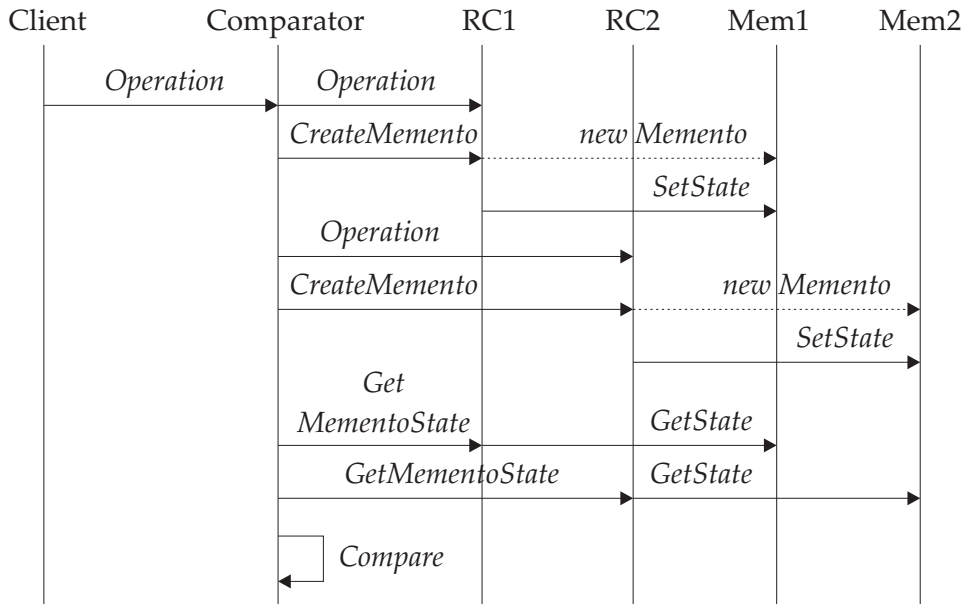
  Perform operations on behalf of clients. Each Recoverable Component is a member of a pair.

- Comparator

  A Proxy [GoF] for a pair of Recoverable Components. The Caretaker for Recoverable Components' Mementos [GoF]. Checks Mementos created by the members of its pair of Recoverable Components. If the Mementos do not match, the Comparator concludes that a fault has occurred in one of its Recoverable Components and initiates corrective action. In systems consisting of two or more pairs, the usual corrective action is to take the faulted pair offline.

Collaborations

- Comparator responds to requests for operations.
- Comparator routes each request to both Recoverable Components, each of which creates a Memento externalizing its state upon completion of the operation.
- Comparator retrieves state from both Mementos and compares them.
- If the states of the Mementos match, Comparator returns the operation's result to the client; otherwise (if the states do not match), Comparator initiates recovery actions.



Consequences

Use of the Comparator-Checked Fault-Tolerant System pattern:

- Improves system tolerance of component faults.
- Substantially increases component costs.
- Increases system complexity. Creating the Memento may require the creation of work queues or other transaction management constructs to ensure consistency of the state data stored in the Memento. Creating the Comparator and its recovery function will also add complexity.
- May impair system latency or throughput if creation of a checkpoint requires processing to pause or stop.

Implementation

- The Comparator's error checking mechanism works by comparing the two Mementos. If the state comparison shows any difference, the pair is taken offline. In some implementations, the ''failed'' pair continues processing inputs but presents no outputs. Continued processing allows the next collaboration.
- The Comparator of a failed pair may collaborate with the error checking mechanisms of the surviving pair's Comparator to identify which Recoverable Component of the failed pair has actually failed. This function can be used to guide manual or automatic intervention, correction, and restart.
- A Comparator may use its Mementos to maintain a consistent externalized image of the ''correct'' state. This can be used to enable the restart of a failed element or its replacement.

Known Uses

The Tandem Nonstop operating system is an example of the Comparator-Checked Fault-Tolerant System pattern.

Related Patterns

Comparator is a Proxy [GoF] and the Caretaker for the Mementos [GoF] of its Recoverable Components.