7.1 Checkpointed System

Intent

Structure a system so that its state can be recovered and restored to a known valid state in case a component fails.

Also Known As

Snapshot, Undo

Motivation

A component failure can result in loss or corruption of state information maintained by the failed component. Systems which rely on retained state for correct operation must be able to recover from loss or corruption of state information.

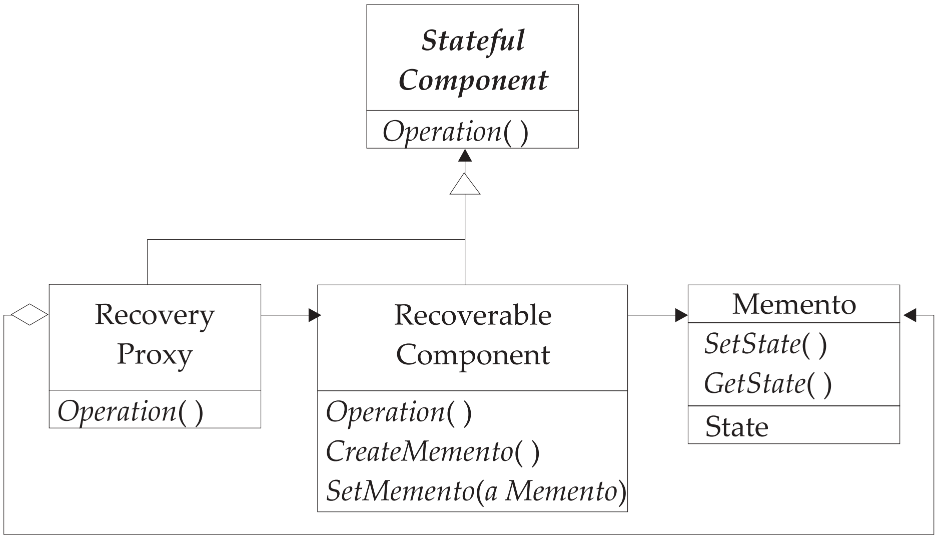
Applicability

Use Checkpointed System when:

- Operations on a component update its state.
- Correctness of the system's operation depends on correctness of its components' state.
- Component failures could cause loss or corruption of a component's state.
- Transactions which occurred between the time a state snapshot is taken and the time the system is rolled back to the snapshot state are irrelevant or inconsequential, or can be reapplied.

Structure

The Checkpointed System pattern consists of a Recovery Proxy [Proxy: GoF] and a Recoverable Component which periodically saves a recoverable version of the component's state as a Memento [GoF]. The Memento can be used to restore the component's state when required.



Participants

- Stateful Component

  Abstract class. Defines component operations.

- Recovery Proxy

  Proxy [GoF] for Recoverable Component. A Stateful Component. Caretaker for Recoverable Component's Mementos. Initiates creation of Mementos when Recoverable Component state changes. Detects failures and initiates state recovery by instructing Recoverable Component to restore state from Memento.

- Recoverable Component

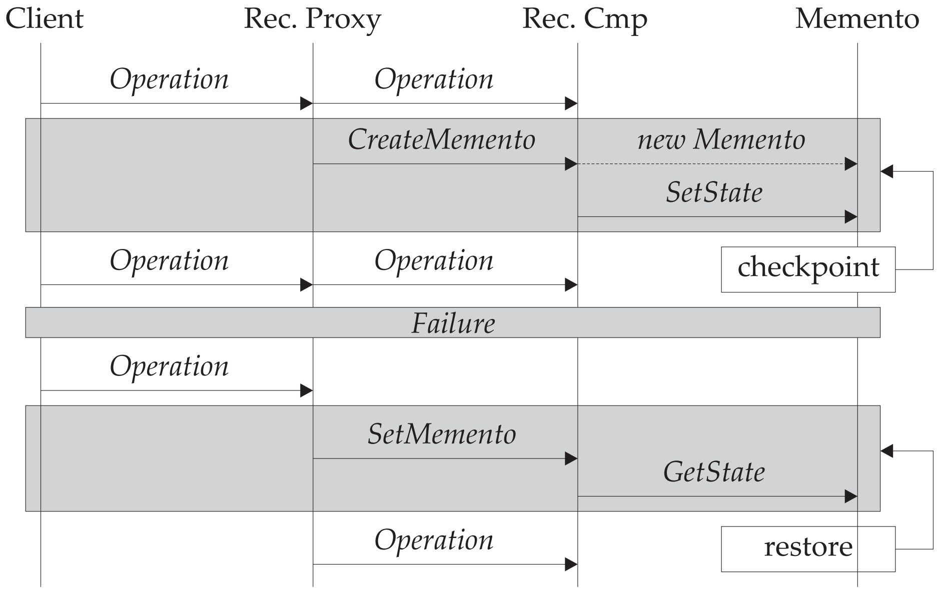
  A Stateful Component. Implements component operations. Periodically saves component state to Memento to support later recovery operations. Restores component state when required.

- Memento [GoF]

  The Recoverable Component's externalized state.

Collaborations

- The Recovery Proxy responds to requests to perform operations.
- The Recovery Proxy periodically instructs the Recoverable Component to create a new Memento to save the Recoverable Component's current state.
- In the event of a failure, the Recovery Proxy instructs the Recoverable Component to restore its state using the information stored in the Memento, and then instructs the Recoverable Component to execute requested operations. Note that any state resulting from operations performed after the most recent state save will be lost.



Consequences

Use of the Checkpointed System pattern:

- Improves component fault tolerance.
- Improves component error recovery.
- Increases system resource consumption (extra resources are required for the Memento).
- Increases system complexity; creating a Memento may require the creation of work queues or other transaction management constructs to ensure consistency of the state data stored in the Memento.
- May increase system latency or decrease throughput if creation of the Memento requires processing to pause or stop.
- Allows loss of a small number of transactions and their associated state.
- Increases system cost per unit of functionality.

Implementation

A wide variety of implementation approaches are possible. Examples include:

- A wide variety of configurations that provide the ability to ‘‘restart’’ the system from a known valid state, either on the same platform or on different platforms.

Known Uses

The periodic save feature of many applications (for example, Microsoft Word) is an instance of the Checkpointed System pattern.

Related Patterns

Recovery Proxy is a Proxy [GoF].
Recovery Proxy is the Caretaker for a Memento [GoF].