**CONTROLLO DEL FLUSSO GLOBALE DEL SOFTWARE**

Il controllo del flusso globale nel sistema CarZone è basato sull'approccio **event-driven**, un meccanismo efficace per gestire le interazioni multiutente e la sequenza delle operazioni. Questo approccio è stato scelto per la sua maturità, la semplicità nella gestione delle operazioni distribuite e la compatibilità con un design modulare orientato agli oggetti.

**1. Definizione e Meccanismi di Controllo del Flusso**

Il controllo del flusso definisce l'ordine delle azioni nel sistema:

**Event-Driven Control** (selezionato per CarZone):

* + Una *main loop* raccoglie eventi esterni (ad esempio input dell'utente o notifiche di sistema) e li distribuisce ai gestori appropriati.
  + Centralizza il controllo del flusso in un'unica struttura, semplificando il design e la gestione.

**2. Implementazione di Event-Driven Control in CarZone**

**Composizione del Sistema**

CarZone utilizza i seguenti componenti per il controllo del flusso basato sugli eventi:

1. **EventStream**:
   * Gestisce un flusso continuo di eventi generati dagli attori (es. *"Visualizza catalogo"*, *"Conferma ordine"*).
2. **Dispatcher**:
   * Centralizza l'instradamento degli eventi, assegnandoli ai gestori appropriati in base al tipo di evento.
3. **Event Handlers**:
   * Oggetti responsabili dell'elaborazione di eventi specifici.
4. **Boundary Objects**:
   * Componenti che mediano l’interazione tra gli attori e il sistema.
5. **Entity Objects**:
   * Contengono i dati persistenti e implementano le operazioni principali associate agli eventi.

**Flusso Operativo**

1. Un attore (Cliente, Utente Non Registrato o Amministratore) genera un evento interagendo con l’interfaccia utente.
2. L'evento è aggiunto all'EventStream.
3. Il Dispatcher analizza l'evento e identifica il gestore appropriato.
4. L'Event Handler esegue le operazioni richieste, interagendo con i sottosistemi coinvolti.

**3. Applicazione nei Sottosistemi**

**Gestione Autovetture**

* **Eventi gestiti**:
  + *"Visualizza catalogo"* (Cliente, Utente Non Registrato).
  + *"Aggiungi auto"*, *"Modifica auto"*, *"Elimina auto"* (Amministratore).
* **Esempio di Flusso**:
  + Il Cliente seleziona *"Visualizza catalogo"*. L'evento viene aggiunto all'EventStream.
  + Il Dispatcher lo inoltra al CatalogEventHandler.
  + Il gestore recupera i dati dal database e restituisce il catalogo aggiornato.

**Gestione Ordini**

* **Eventi gestiti**:
  + *"Visualizza ordini"* (Cliente, Amministratore).
  + *"Conferma ordine"* (Cliente).
  + *"Approva ordine"* (Amministratore).
* **Esempio di Flusso**:
  + Un Cliente seleziona *"Conferma ordine"*. L'evento è aggiunto all'EventStream.
  + Il Dispatcher invia l'evento al OrderEventHandler.
  + Il gestore valida l'ordine, aggiorna il database e restituisce una conferma.

**4. Benefici del Controllo Event-Driven**

1. **Modularità**:
   * La logica è suddivisa in gestori dedicati, ciascuno focalizzato su una specifica funzionalità.
2. **Centralizzazione**:
   * Il Dispatcher centralizza il controllo degli eventi, semplificando il monitoraggio e la gestione.
3. **Scalabilità**:
   * Nuovi eventi possono essere aggiunti con modifiche minime al sistema.
4. **Manutenzione**:
   * La localizzazione del controllo in gestori specifici rende il sistema più semplice da aggiornare o estendere.