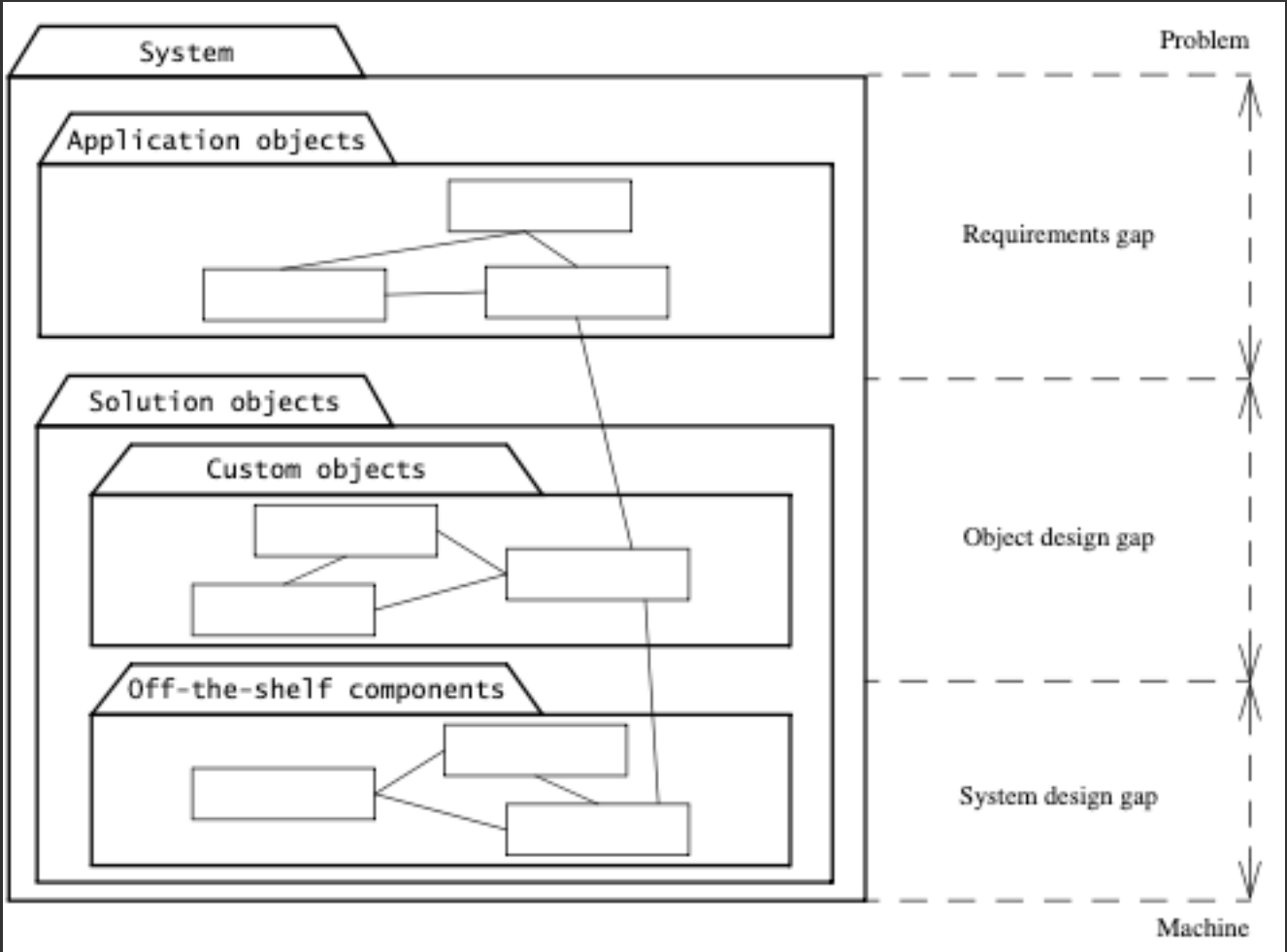


L’**analisi** riduce il divario tra il problema e la macchina identificando gli oggetti che rappresentano concetti specifici del problema. Durante tale fase il sistema viene descritto in termini di comportamento esterno come la sua funzionalità(modello del caso d’uso), i concetti del domino dell’applicazione che manipola(modello a oggetti), il suo comportamento in termini di interazioni (modello dinamico) e i suoi requisiti non funzionali.

Il **sistem design** invece riduce il divario tra il problema e la macchina in due modi:

1. Tale progettazione genera una macchina virtuale che offre un livello di astrazione più elevato rispetto alla macchina tramite l’utilizzo di middleware,toolkit.
2. Tale progettazione identifica i componenti off-the-shell per gli oggetti del dominio dell’applicazione come librerie di classi.

Alla fine di tutto, rimane un puzzle con alcuni pezzi mancanti. Tali pezzi lo si trovano nella fase di **Object Design** includendo l’identificazione di nuovi oggetti soluzione, l’adattamento di componenti off-the-shell e la specifica precisa di ogni interfaccia e classe di sottosistema. Il modello di tale progettazione può essere suddiviso in gruppi che possono essere implementati dai singoli sviluppatori.



Object Design chiude il gap tra oggetti applicazione identificati durante l’analisi e i componenti off-the-shell scelti durante il sistem design.

L’Object Design include:

- **Riutilizzo/Riuso:** i componenti off-the-shell identificare durante il sistem design sono utilizzate nella realizzazione di ogni sottosistema. Librerie di classi e componenti aggiuntivi sono selezionati per strutture e servizi di dati di base. Vengono selezionati dei Design Pattern per risolvere problemi comuni e per proteggere classi specifiche da cambiamenti futuri. Spesso i componenti e i Design Pattern devono essere adorati prima di poter essere utilizzati. Ciò viene fatto:
 - attraverso oggetti rapper, quindi avvolgendo oggetti personalizzati attorno ad essi.
 - Raffinandoli utilizzando l’ereditarietà.
- **Specifica dell’interfaccia/servizi:** i servizi del sottosistema identificati durante la progettazione del sistema sono specificati in termini di interfacce di classe, tra cui operazioni, argomenti, firme dei tipi ed eccezioni. Sono identificati anche ulteriori operazioni ed oggetti necessari per trasferire dati tra i sottosistemi. Il risultato di tale fase è una specifica di interfaccia completa per ciascun sottosistema. Tale specifica viene spesso chiamata API del sottosistema.
- **Ristrutturazione del modello a oggetti:** manipola il modello di sistema per aumentare il riutilizzo del codice o soddisfare altri design goal. Durante la ristrutturazione ci si occupa anche di come soddisfare design goal come mantenimento, leggibilità e comprensione del modello di sistema. Attività tipiche di ristrutturazione sono:
 - Trasformare associazioni N-arie in binarie.
 - Implementare associazioni binarie attraverso riferimenti.
 - Fondere classi simili in differenti sottosistemi in un’unica classe.
 - Trasformare classi con nessun comportamento in attributi.
 - Decomporre classi complesse in classi più semplici.
 - Aumentare l’ereditarietà ed il packaging modificando classi ed operazioni.
- **Ottimizzazione del modello ad oggetti:** si occupa ai requisiti non funzionali del modello di sistema. Ovvero:
 - Modificare gli algoritmi per rispondere ai requisiti di velocità o di memoria.
 - Ridurre la molteplicità nelle associazioni per accelerare le query.
 - L’aggiunta di associazioni ridondanti per l’efficienze.
 - Modificare l’ordine di esecuzione.
 - Aggiungere attributi derivati per migliorare il tempo di accesso agli oggetti.
 - Aprire l’architettura ovvero aggiungere la possibilità di accedere a strati di basso livello)

Tali attività appena descritti non vengono eseguiti in una maniera sequenziale bensì in un modo concorrente. Ci potrebbero però essere delle dipendenze; uno specifico componente off-the-shell può limitare il numero di tipi di eccezioni menzionati nella specifica di un’operazione e quindi avere un impatto sull’interfaccia del sottosistema. La ristrutturazione e l’ottimizzazione possono ridurre il numero di oggetti da implementare e quindi aumentare il riuso.

Di solito, le specifiche dell’interfaccia e l’attività di riuso vengono realizzati per primi producendo un modello di progettazione di oggetti che viene verificato rispetto ai casi d’uso che esercitano il sottosistema specifico. Una volta che il modello si è stabilizzato vengono svolte le attività di ristrutturazione ed ottimizzazione.