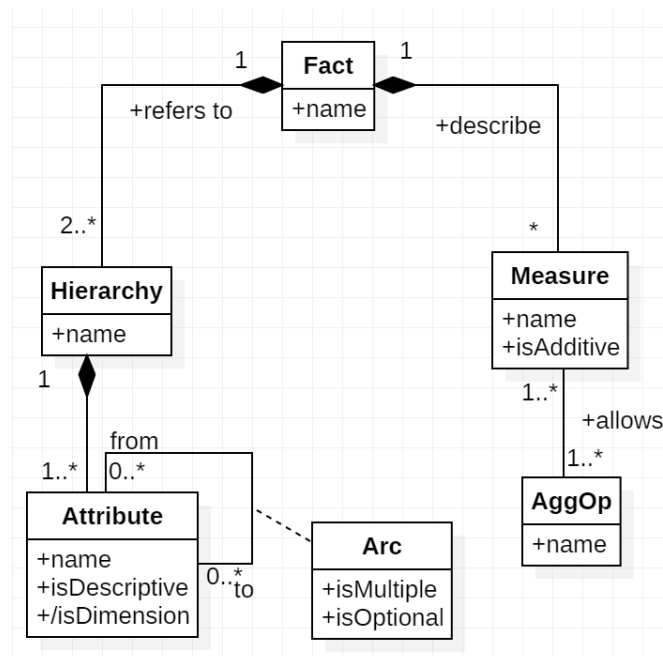
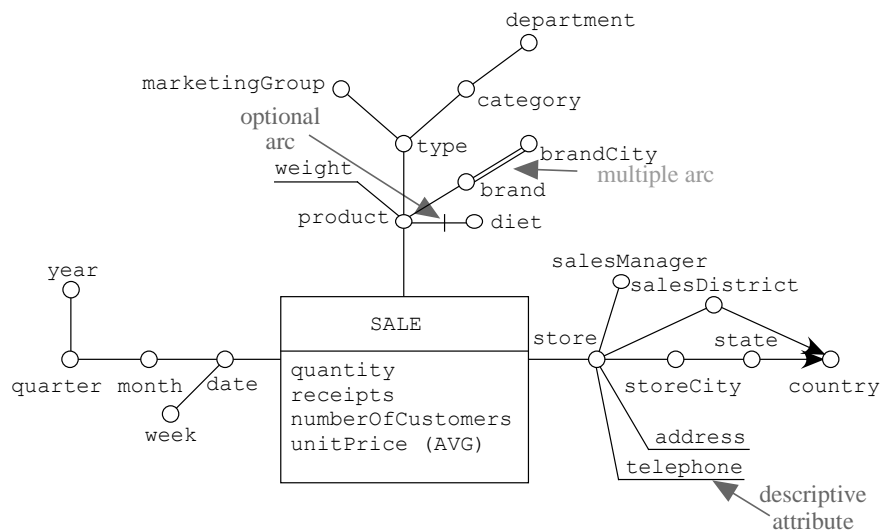
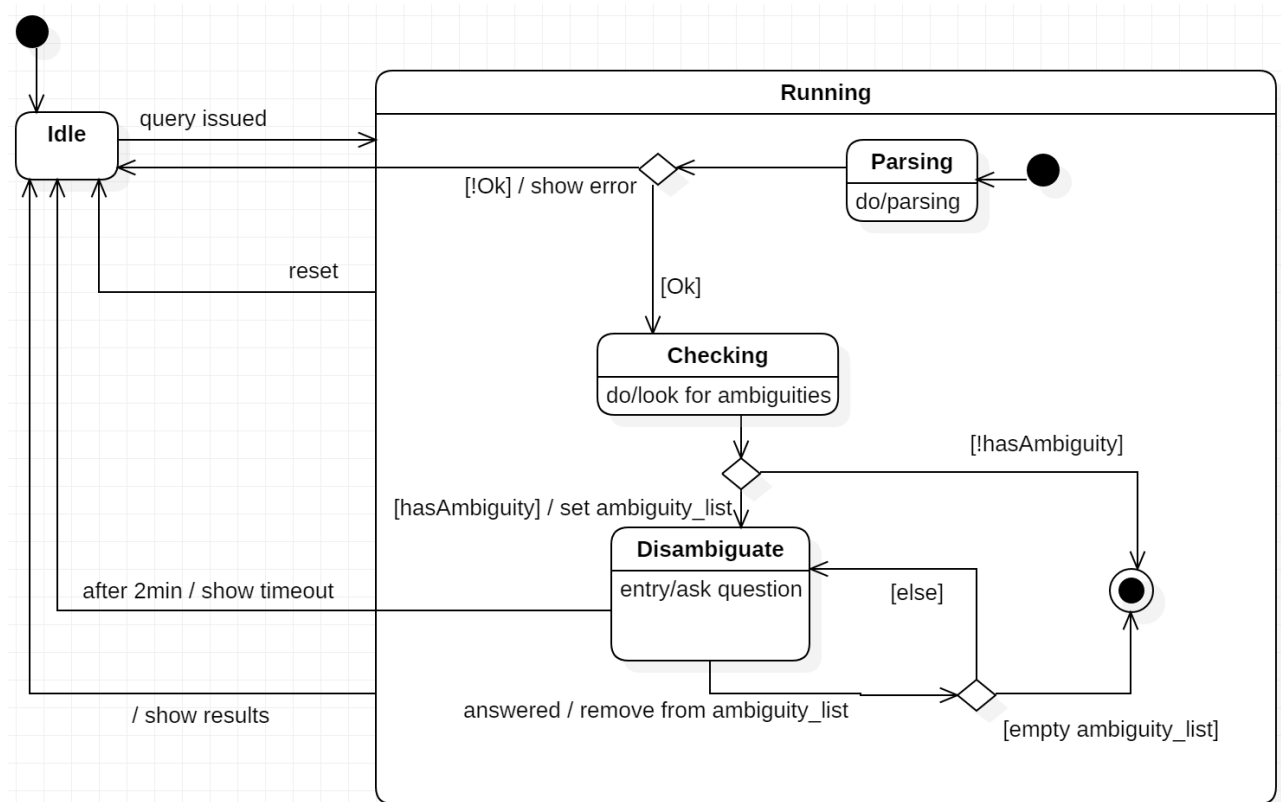


1) Si vuole meta-modellare con un diagramma delle classi una versione semplificata del Dimensional Fact Model (DFM), un formalismo grafico per la modellazione concettuale di Data Warehouse. Il DFM si sviluppa attorno al concetto di fatto, ovvero un fenomeno di business (ad esempio le vendite di prodotti, *SALE* in figura). Ogni fatto include un insieme (anche vuoto) di misure (ad esempio la quantità di prodotti venduta, *quantity*, e l'incasso, *receipts*). A ciascuna misura è associato almeno un operatore di aggregazione (ad esempio somma o media); una misura si dice additiva se è associata alla somma, non additiva in caso contrario. Un fatto è caratterizzato da due o più gerarchie, ciascuna strutturata come un grafo di attributi (mostrati in figura come cerchietti). Tra gli archi di questi grafi, alcuni possono essere multipli e/o opzionali (v. figura). Tra gli attributi, alcuni possono essere descrittivi (per esempio *address*). In ogni gerarchia, uno degli attributi è graficamente attaccato al fatto ed è detto dimensione (*store*, *product* e *date* in figura) (*11/32 punti*).



2) Si modelli con diagramma degli stati il funzionamento di COOL, un'applicazione a supporto della traduzione in SQL di query in linguaggio naturale. All'avvio di COOL, il sistema attende una prima query dell'utente. Una volta ricevuta la query, viene avviata l'interpretazione della query, che si compone di due fasi: parsing e verifica della query. Se l'interpretazione va a buon fine, COOL restituisce il risultato della query all'utente e si mette in attesa di nuove query. Altrimenti, nel caso in cui il parsing fallisca, il sistema ritorna in fase di attesa e l'errore è segnalato all'utente. Se in fase di verifica COOL riscontra l'esistenza di ambiguità, il sistema entra in fase di disambiguazione. Per ogni ambiguità, il sistema pone una domanda all'utente e ne attende la risposta. Una volta esaurite le domande, il sistema conclude l'interpretazione. Se l'utente non risponde alla domanda entro due minuti, il sistema ritorna automaticamente in fase di attesa segnalando un errore di "timeout". In ogni momento il comando reset ripristina COOL in stato di attesa di una nuova query (9/32 punti).



3) Si spieghi la differenza tra testing in the small e testing in the large, portando anche alcuni esempi (6/32 punti).

4) Si illustrino i benefici dell'approccio orientato agli oggetti nella progettazione del software, citando i problemi presenti nell'approccio funzionale (6/32 punti).