
Analisi e progettazione del software

Studio di caso: **ERedit**

20 marzo 2018

Acme ERedit (nel seguito chiamato semplicemente **ERedit**) è un sistema software, di tipo client-server, usato dall'azienda *Acme Software Corporation* la gestione dei propri diagrammi Entità-Relazione (E-R).

Il sistema ERedit consente di gestire e rappresentare:

- I propri utenti, che sono i dipendenti della *Acme Software Corporation* responsabili della progettazione di basi di dati. Ad esempio, *Mario Rossi*.
- Un insieme di diagrammi E-R, ciascuno dei quali ha un nome che lo identifica ed una descrizione. Ad esempio, il diagramma *Università*, con descrizione *Schema dell'Università Acme*. Ogni diagramma è stato creato da un utente del sistema. Un utente può accedere solo ai diagrammi E-R che ha creato.
- Un diagramma E-R può contenere delle entità, ciascuna con un nome (che identifica l'entità nell'ambito di un diagramma) ed una descrizione. Ad esempio, l'entità *Studente*, con descrizione *Studente di questa università*.
- Ad un'entità possono essere associati degli attributi, ciascuno con un nome (che lo identifica nell'ambito dell'entità), un tipo ed una descrizione. Ad esempio, l'attributo *Nome* di *Studente*, di tipo *String*, che è *Nome di questo studente*.
- Un diagramma E-R può contenere delle relazioni binarie (ovvero, relazioni a cui partecipano due entità). Ciascuna relazione ha un nome (che la identifica nell'ambito di un diagramma) ed una descrizione. Ad esempio, la relazione *Iscrizione*, tra *Studente* e *Corso di studi*, con descrizione *Il corso di studi a cui è attualmente iscritto uno studente*. Sono ammesse anche relazioni binarie ricorsive, che coinvolgono due volte la stessa entità.
- Nota: Non sono di interesse, al momento, né identificatori delle entità, né relazioni N-arie, né attributi delle relazioni, né cardinalità delle relazioni, né generalizzazioni-specializzazioni. Non sono di interesse nemmeno versioni diverse di uno stesso schema, e nemmeno la suddivisione di un singolo schema in più "viste".

Il sistema ERedit consente di creare, editare e modificare diagrammi E-R. Normalmente, un utente utilizza il sistema in questo modo:

- per prima cosa effettua il login (operazione di sistema 1);
- poi crea un nuovo diagramma (operazione di sistema 2) oppure apre un diagramma esistente (operazione di sistema 3);
- quindi effettua l'editing del diagramma corrente, eseguendo (in ordine qualsiasi, e un numero di volte qualsiasi) le operazioni di sistema 4-8;
- infine, chiude il diagramma che ha editato (operazione di sistema 9).

In particolare, dall'analisi dei casi d'uso del sistema ERedit, sono state identificate le seguenti operazioni di sistema:

1. Login di un Utente. L'Utente inserisce il suo indirizzo di posta elettronica e la sua password. Il Sistema verifica la correttezza dei dati immessi, e autentica l'Utente. Il Sistema mostra nome e cognome dell'Utente. Da quel momento in poi, l'Utente è considerato l'utente corrente. Il Sistema mostra l'elenco di tutti i diagrammi creati da quell'utente.
2. Creazione di un nuovo diagramma E-R, dato il nome e la descrizione del diagramma. Il diagramma è inizialmente vuoto. L'utente corrente è considerato il creatore del diagramma. Da quel momento in poi, il diagramma creato è considerato il diagramma corrente.
3. Apertura di un diagramma E-R, dato il nome del diagramma. Il diagramma viene cercato tra i diagrammi dell'utente corrente, caricato e visualizzato: vengono visualizzate tutte le sue entità (ciascuna con i suoi attributi) e tutte le sue relazioni (ciascuna con l'indicazione delle entità che vi partecipano). Da quel momento in poi, il diagramma aperto è considerato il diagramma corrente.
4. Inserimento di una nuova entità nel diagramma corrente, dato il nome e la descrizione dell'entità.
5. Inserimento di una nuova relazione binaria nel diagramma corrente, dato il nome e la descrizione della relazione, nonché i nomi delle due entità coinvolte dalla relazione.
6. Inserimento di un nuovo attributo in una entità del diagramma corrente, dato il nome, il tipo e la descrizione dell'attributo, nonché il nome dell'entità a cui l'attributo va associato.
7. Cancellazione di una relazione dal diagramma corrente, dato il suo nome.
8. Cancellazione di una entità dal diagramma corrente, dato il suo nome. Insieme all'entità vanno cancellati anche tutti i suoi attributi, nonché tutte le relazioni a cui l'entità partecipa.
9. Chiusura del diagramma corrente (che cessa di essere corrente).

Analisi e progettazione del software

Studio di caso: ERedit

Esercizio A1 (ANALISI)

Fare l'analisi orientata agli oggetti per il sistema in discussione, relativamente a tutte le operazioni di sistema mostrate, come segue:

- Mostrare il modello di dominio.
- Mostrare un diagramma di oggetti di dominio che rappresenta:
 - gli utenti *Mario Rossi* e *Paolo Bianchi*;
 - i diagrammi E-R *Biblioteca* e *Università* creati da *Mario Rossi*;
 - in particolare, questo diagramma *Università* contiene:
 - l'entità *Studente*, con attributi *nome*, *cognome* e *matricola*;
 - l'entità *Corso*, con attributi *nome*, *codice* e *cfu*;
 - la relazione *Frequenza*, tra l'entità *Studente* e l'entità *Corso*;
 - la relazione *Propedeuticità*, tra l'entità *Corso* e l'entità *Corso*.

Esercizio A2 (ANALISI)

Fare l'analisi orientata agli oggetti per il sistema in discussione, relativamente a tutte le operazioni di sistema mostrate, come segue:

- Mostrare il contratto di tutte le operazioni di sistema mostrate.

Ipotesi di lavoro, valide per tutti gli esercizi di progettazione.

- In tutti gli esercizi che seguono, si faccia l'ipotesi che il sistema in discussione gestisca i propri dati solo in memoria principale. Si supponga anche che durante il caso d'uso di avviamento vengano creati e caricati in memoria tutti gli oggetti le cui informazioni siano già effettivamente disponibili al momento dell'avviamento.
- Per ciascuna operazione di sistema va creato un diagramma di interazione che descrive l'interazione relativa alla trasformazione (cambiamento di stato) provocata dall'operazione di sistema. Per quanto riguarda invece le risposte (interrogazioni) restituite dal sistema, se nessun esercizio lo richiede esplicitamente allora non bisogna mostrare nei diagrammi di interazione né il calcolo dei dati da restituire né la loro visualizzazione. Tuttavia, per le risposte del sistema, è comunque necessario verificare che i dati da restituire possano essere (facilmente) calcolati sulla base delle navigabilità tra gli oggetti che sono state progettate (vedi anche il punto successivo).
- **Le soluzioni individuate dovranno essere compatibili (in particolare in termini di visibilità, ovvero di navigabilità delle associazioni) con la realizzazione di TUTTE le operazioni di TUTTI i casi d'uso descritti tra i requisiti.**
- **Nei diagrammi di interazione, mostrare IN MODO ESPLICITO: tutti gli OGGETTI che partecipano all'interazione (compresi gli OGGETTI COLLEZIONE), tutti i MESSAGGI scambiati tra oggetti, tutte le CREAZIONI di oggetti e tutte le FORMAZIONI e ROTTURE di COLLEGAMENTI.**
- Nei diagrammi di interazione, motivare le scelte di progetto fatte indicando i pattern GRASP e GoF applicati.
- Nei diagrammi delle classi di progetto, mostrare: (1) per ciascuna classe: il nome della classe, i nomi dei suoi attributi, i nomi delle sue operazioni; e (2) per ciascuna associazione e ciascuna sua estremità navigabile: la freccia di navigabilità, il nome dell'estremità, la molteplicità e, in caso di associazione navigabile a molti, il tipo di collezione scelta.

Esercizio A3 (PROGETTAZIONE)

Fare la progettazione orientata agli oggetti per il sistema in discussione, relativamente a tutte le operazioni di sistema mostrate, come segue:

- Mostrare i diagrammi di interazione relativi a tutte le operazioni di sistema mostrate.
- Mostrare il corrispondente diagramma delle classi di progetto.