

IV. Manuale d'uso del prototipo di modellatore UML

In questo capitolo si illustra brevemente l'uso del prototipo di modellatore UML che è stato realizzato a seguito della progettazione finora illustrata.

4.1 Funzionalità

Allo stato attuale questo prototipo permette all'utente di creare un diagramma delle classi ed il relativo modello UML attraverso un'interfaccia semplice ed intuitiva. L'area di disegno è predisposta a gestire più diagrammi contemporaneamente, ma al momento si può disegnare su un unico diagramma. Le operazioni di disegno sono semplificate dalla presenza di una barra di bottoni che permette la creazione degli elementi del diagramma. Ogni bottone è caratterizzato da un'icona grafica che permette all'utente di capire in modo spontaneo quale funzione può compiere; inoltre un utente può posizionare per alcuni istanti il cursore del mouse su un bottone che non conosce ed avere in risposta una breve nota che specifica a cosa serve.

Le operazioni di salvataggio e ripristino di un diagramma non sono ancora implementate, comunque saranno rese disponibili in breve tempo.

4.2 Finestra principale

Si presenta come una consueta applicazione dell'ambiente X-Window, alla sommità vi è una barra di menù, sotto una barra per i bottoni ed al centro vi è la finestra per il disegno dei diagrammi di classe.

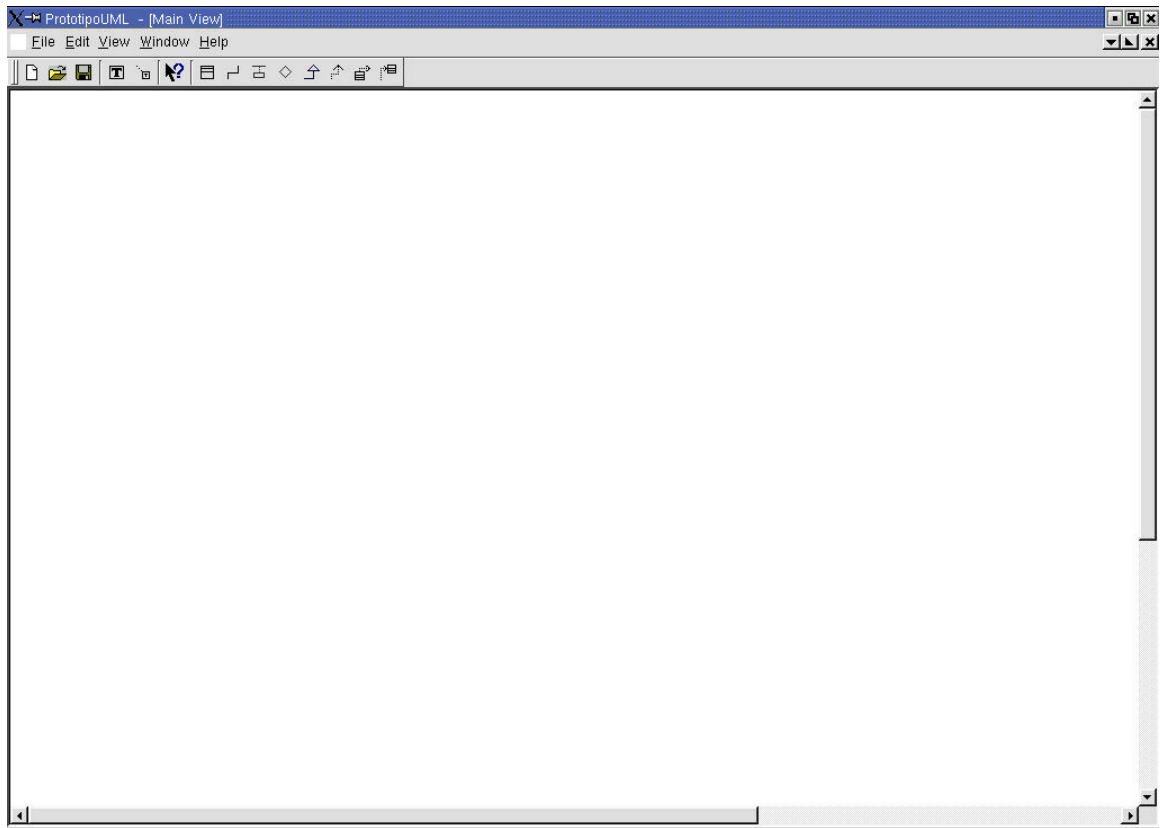


Figura 4.1 – Prototipo di modellatore UML.

La barra di menù contiene gli usuali controlli per la creazione, la modifica, il salvataggio/ripristino di diagrammi di classe e per la modifica e visualizzazione degli stessi.

I primi tre bottoni della relativa barra sono rispettivamente: *New diagram*, *Open diagram* e *Save diagram*, ovvero crea un nuovo diagramma, apri e salva diagramma. Il quarto ed il quinto sono: *Text note* ed *Anchor text note* (non ancora implementati), rispettivamente crea un’etichetta di testo e collega l’etichetta ad un elemento del diagramma.

Ora si introduce l’uso dei bottoni relativi alla creazione dei diagrammi di classe con un piccolo esempio di modello.

4.2.1 Creazione del diagramma di classi di una biblioteca

Se si vuole creare una classe che rappresenti gli oggetti tipo libro (v. esempio pag. 21), allora si fa click con il mouse sul bottone *New Class*.

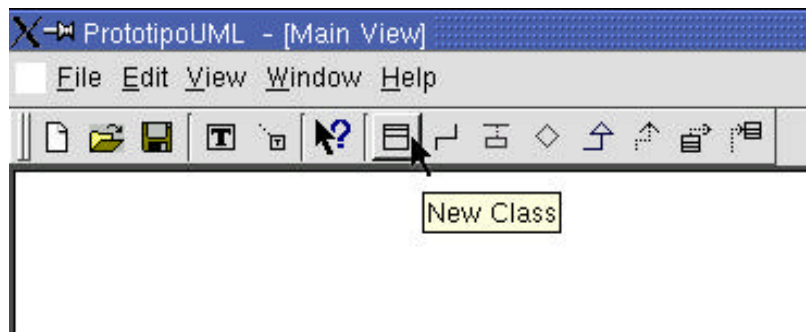


Figura 4.2 – La barra dei bottoni.

Ciò fatto compare un rettangolo con tre scompartimenti vuoti in alto a sinistra del diagramma delle classi; questo rappresenta una nuova classe nel modello corrispondente al diagramma che si sta disegnando e al momento ha un nome non specificato e non possiede alcun attributo od operazione.

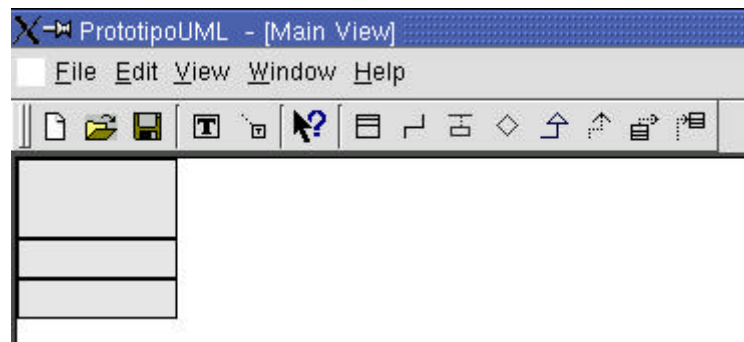


Figura 4.3 – Una nuova classe nel diagramma.

Per spostare la nuova classe sul diagramma basta puntare il cursore del mouse su qualsiasi punto del rettangolo e tenere premuto il tasto sinistro del mouse; la classe sparisce mentre si mantiene premuto il tasto ed al suo posto compare il suo

contorno tratteggiato; questo segue i movimenti del mouse fintantoché è premuto il tasto sinistro del mouse. Al rilascio del tasto la classe torna a prendere il posto del contorno e quindi si posiziona nella zona voluta del diagramma.

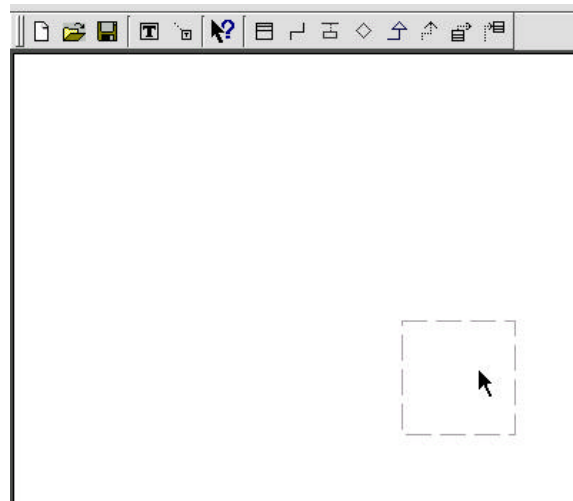


Figura 4.4 – Come spostare una classe.

Inoltre, se su di un oggetto del diagramma viene premuto il tasto destro del mouse compare il menù contestuale relativo alle operazioni possibili su tale oggetto.

Se si vuole cambiare il nome della classe in “Libro”, si sceglie quindi il controllo *Rename* dal menù contestuale.

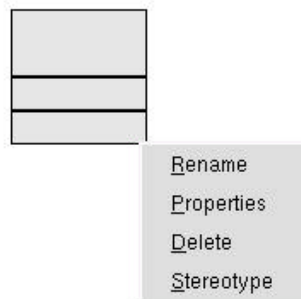


Figura 4.5 – Il menù contestuale per una classe.

A questo punto compare la finestra di dialogo che permette di cambiare il nome a questa classe di nome predefinito “*Unnamed_Classifier*”.

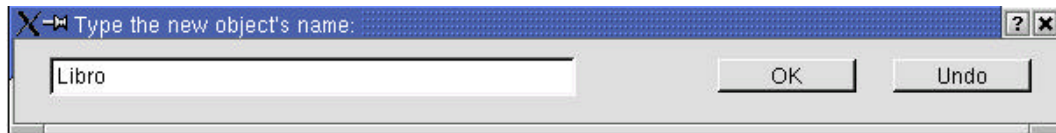


Figura 4.6 – La finestra che rinomina un elemento del diagramma.

Si inserisce il nuovo nome nel campo per l’inserimento di testo e si fa click sul bottone *OK*. Ora compare il nuovo nome nello scompartimento più in alto della classe, centrato rispetto a questo ed in grassetto, come da specifica della notazione UML.

Se si vogliono aggiungere nuove caratteristiche strutturali alla classe si fa comparire di nuovo il menù contestuale relativo alla classe *Libro* e si sceglie la voce *Properties*. Compare allora la finestra delle proprietà della classe che contiene tre pagine di dialogo accessibili con tre segnalibri alla sommità di ciascuna: *Properties*, *Attributes* ed *Operations*.

Se la classe è già stata modificata in precedenza la finestra delle proprietà si modifica in modo da presentare all’utente tutte le caratteristiche della classe e permette il loro aggiornamento. Il nome della classe che si sta modificando appare nel titolo della finestra sulla sua sommità. La prima pagina della finestra riguarda le caratteristiche relative all’intera classe e tali caratteristiche si possono cambiare con alcuni tasti di spunta, combo box, campi per inserimento di testo.

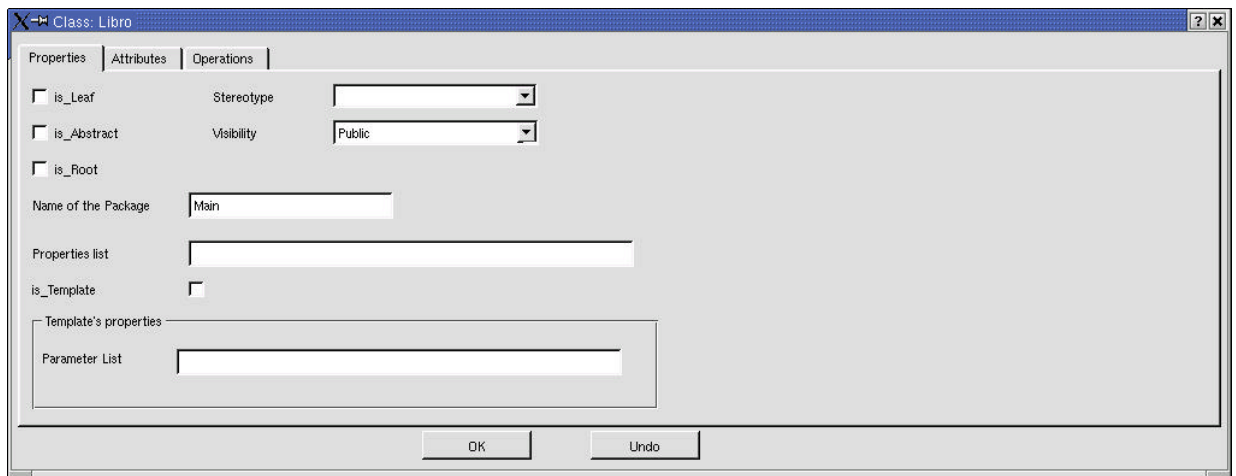


Figura 4.7 – La finestra che cambia le proprietà di una classe.

Se si vogliono aggiungere nuovi attributi bisogna fare click sul segnalibro *Attributes* e compare la seguente pagina di dialogo composta da una tabella le cui righe corrispondono ciascuna ad un attributo della classe: il tasto *Add* aggiunge un nuovo attributo alla classe, il tasto *Delete* elimina, previa conferma, l'attributo che precedentemente è stato selezionato con un click.

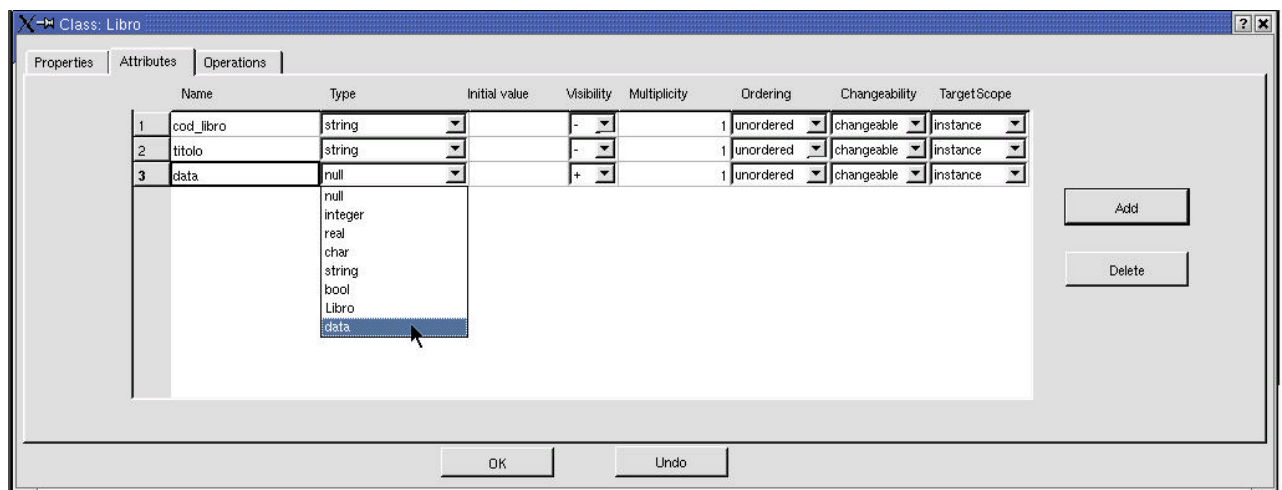


Figura 4.8 – Come aggiungere nuovi attributi ad una classe.

Nella tabella in figura sono già stati aggiunti tre attributi: il campo *Name* è uno spazio di inserimento di testo ove si digita il nome del nuovo attributo o vi

compare il nome dell'attributo già presente nella classe. Il campo *Type* è un combo box che contiene i tipi che si possono dare all'attributo; i tipi sono quelli predefiniti del C++ e quelli che si creano per ogni nuova classe che si aggiunge nel diagramma e quindi al modello. Nella figura 4.8 si vede che è stata premuta la freccia all'estrema destra del combo box del terzo attributo, in modo da fare comparire la lista dei tipi attualmente disponibili.

Nel campo *Initial value* vi si può inserire il valore di default dell'attributo.

Il campo *Visibility* definisce la visibilità dell'attributo ed è codificata secondo la notazione dello UML: '+' sta per *public*, '-' sta per *private*, '#' sta per *protected* e '~' sta per *package*.

Il campo *Multiplicity* contiene la molteplicità dell'attributo che va inserita come una stringa di testo che rispetta le convenzioni discusse nel paragrafo 2.2.1.

Ordering invece dà la possibilità di scegliere tra un valore *ordered* o *unordered* nel caso di molteplicità maggiore di 1.

Changeability dà la possibilità di scegliere fra *changeable*, *frozen* e *addOnly*.

Target Scope dà la possibilità di scegliere tra *instance* e *classifier*.

Tutti i precedenti campi se non modificati prendono il valore predefinito come di default dallo UML.

Se si vogliono aggiungere delle operazioni si fa click sul segnalibro corrispondente a *Operations* e compare la seguente pagina di dialogo con la tabella delle operazioni dal funzionamento simile a quella precedente.

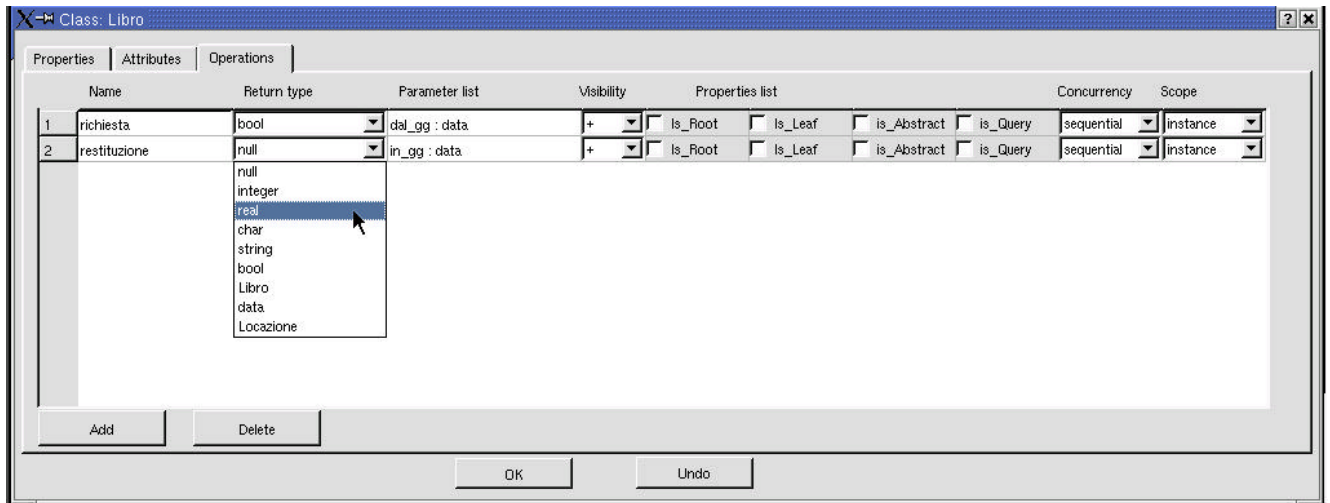


Figura 4.9 – Come aggiungere nuove operazioni ad una classe.

Si illustrano di seguito i campi di ciascuna operazione corrispondente a ciascuna riga della tabella.

- *Name* è il nome dell'operazione.
- *Return type* è il tipo restituito dalla funzione.
- *Parameter list* contiene la lista dei parametri dell'operazione. Questi sono da inserire secondo la sintassi suggerita dalla stessa finestra di dialogo quando si aggiunge una nuova operazione: *namepar: type= default*. Infatti, prima va digitato il nome del parametro, poi il carattere ':', poi il tipo del parametro (deve essere un tipo predefinito del C++ o definito all'interno del modello utente), poi, opzionale, il carattere '=' seguito dal valore opzionale di default dello stesso parametro. Se vi sono altri parametri si possono aggiungere separandoli con il carattere ',' altrimenti se non ve ne sono affatto si cancella la stringa contenuta in questo campo.

- *Properties List* contiene delle celle di spunta corrispondenti ciascuna ad una caratteristica dell'operazione. Se se ne contrassegna una con un click (vi compare una 'v') significa che è stato dato il valore *true* alla caratteristica evidenziata dalla corrispondente etichetta: *is_Root*, *is_Leaf*, *is_Abstract* ed *is_Query*.
- *Concurrency* permette di scegliere fra *sequential*, *guarded* e *concurrent*.
- *Scope* permette di scegliere fra *instance* e *classifier*.

Se a questo punto si fa click su *OK* vengono aggiunte tutte le modifiche fatte al modello e il diagramma ne mostra gli effetti sulla classe.



Figura 4.10 – La classe *Libro*.

Aggiungiamo ora altre classi al diagramma con passi simili ai precedenti.

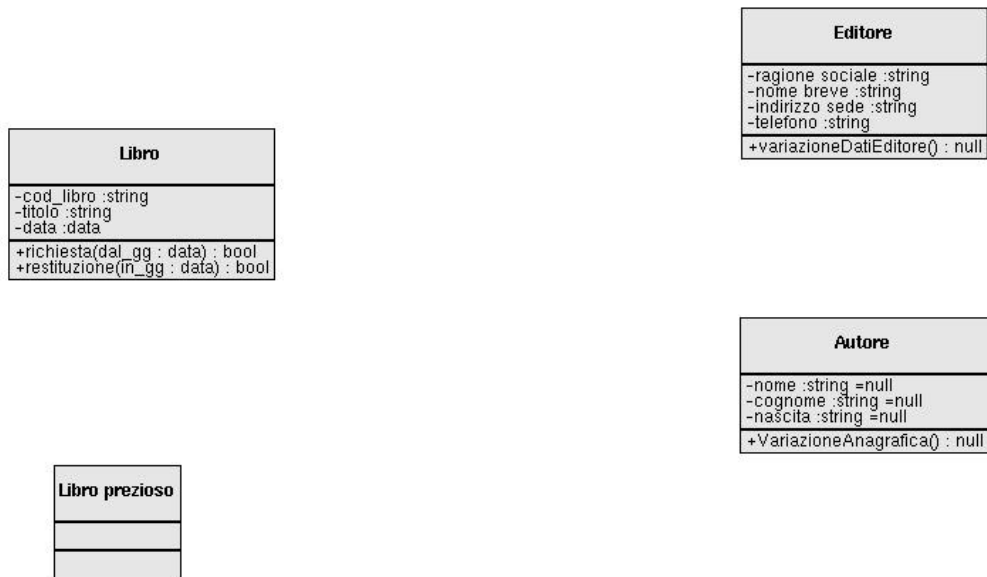



Figura 4.11 – Classi del modello di una biblioteca.

Si vogliono ora creare le associazioni fra queste classi. Per creare un'associazione fra *Libro* ed *Editore* si fa click sul simbolo dell'associazione binaria sulla barra dei bottoni: .

A questo punto il cursore cambia aspetto per informare l'utente che si sta creando un'associazione. In qualsiasi momento si può annullare l'operazione facendo click sul tasto destro del mouse (come del resto in tutte le operazioni che creano una qualsiasi relazione). Per scegliere la partenza dell'associazione si fa click (si rilascia quindi immediatamente il tasto) col tasto sinistro del mouse su un punto qualsiasi della classe prescelta; a questo punto se si muove il mouse si vede una linea tratteggiata in grigio che parte dalla classe prescelta ed unisce in modo continuo il puntatore (si va infatti a disegnare il cammino dalla classe *Libro* a quella *Editore*).

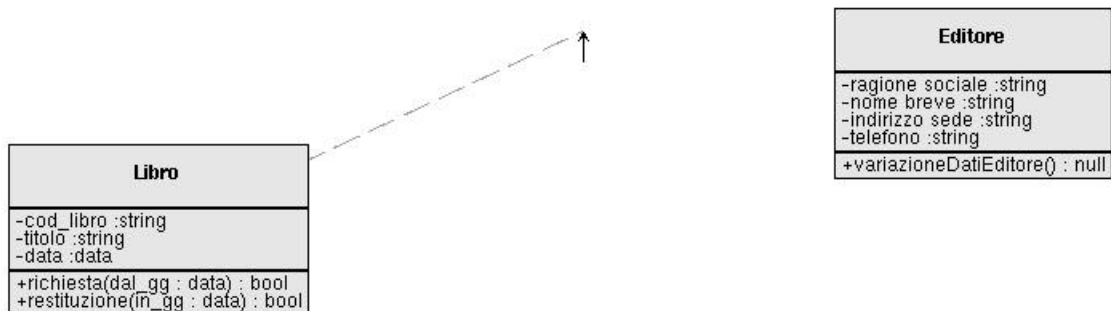


Figura 4.12 – Creazione del cammino di una relazione.

Se si vuole inserire un vertice per spezzare il cammino si fa click nel punto ove si vuole incernierare il cammino e di conseguenza appare un piccolo quadrato che rappresenta il vertice trascinabile del cammino.

Ora se si fa click su un punto qualsiasi della classe editore si forma una nuova associazione nel modello fra Libro ed Editore rappresentata nel diagramma come in figura.

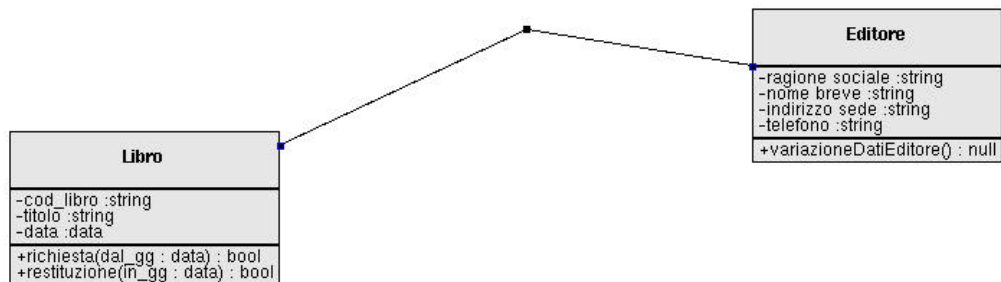


Figura 4.13 – Un'associazione fra due classi.

Se si vuole cambiare la posizione del vertice dell'associazione lo si punta con il cursore del mouse e si tiene premuto il tasto sinistro; come effetto il cammino tra le due classi si colora di grigio e se si muove il mouse si sposta il vertice e di conseguenza anche il cammino.

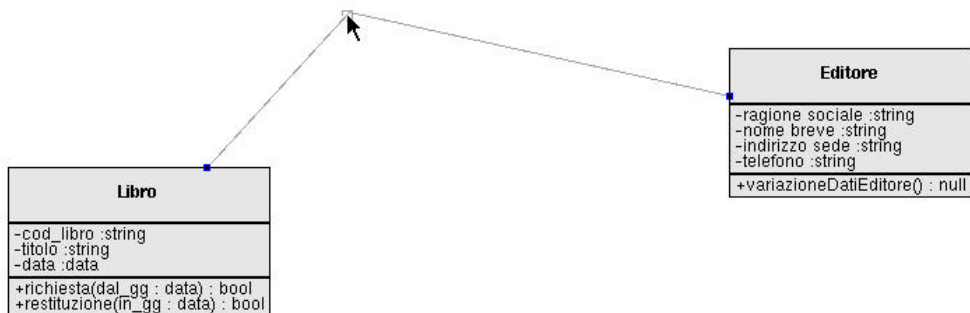


Figura 4.14 - Come spostare un vertice del cammino di un'associazione.

Quando si rilascia il tasto il cammino riprende il colore originale e si fissa nella nuova posizione.

Premendo il tasto destro del mouse quando il puntatore è su un punto qualsiasi dell'associazione compare un menù contestuale simile a quello visto per le classi, che permette di richiamare le finestre di dialogo relative all'associazione.

In questo modo possiamo rinominare l'associazione con il nome *pubblicato da*.

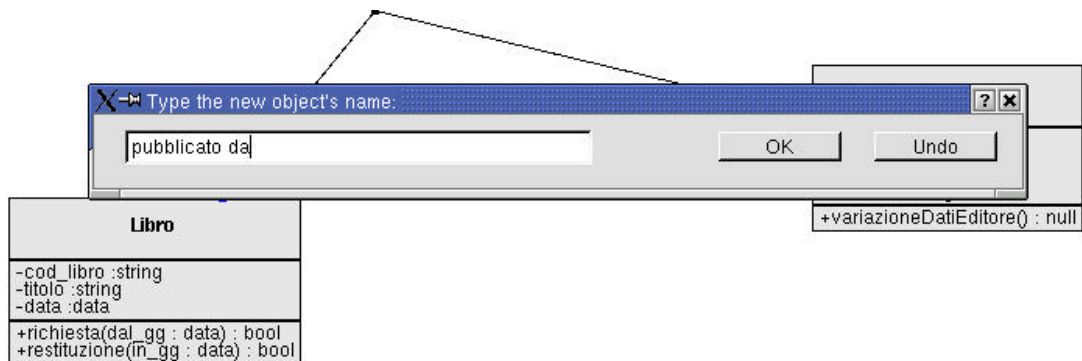


Figura 4.15 – Come cambiare il nome ad un'associazione.

Se invece si sceglie dal menù contestuale la voce *Properties* compare la seguente finestra di dialogo.

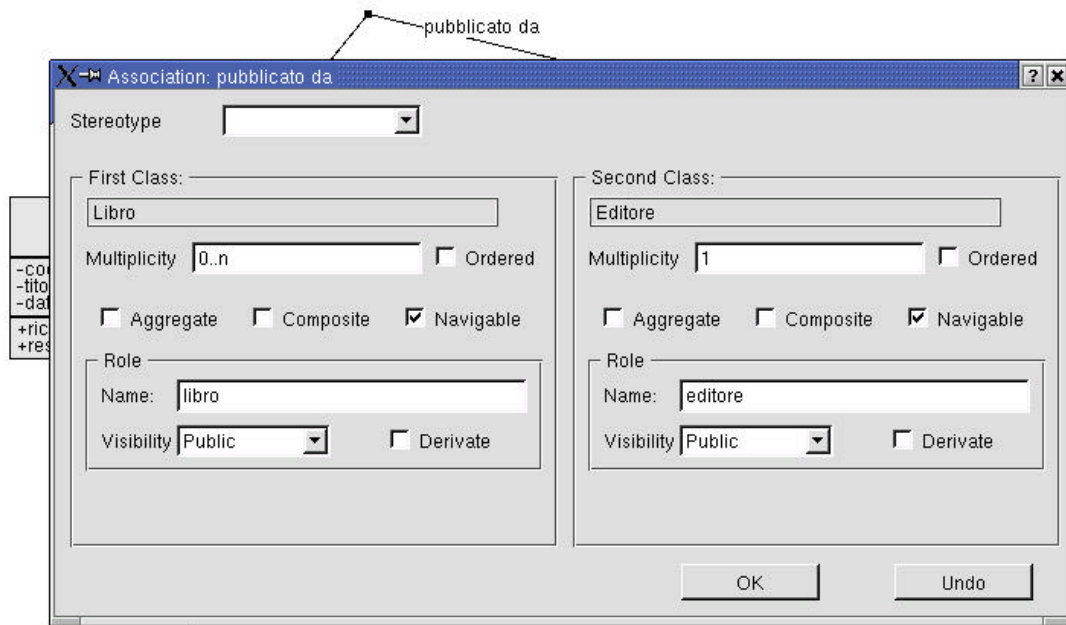


Figura 4.16 – La finestra che cambia le proprietà di un'associazione.

Nella barra di titolo in alto la finestra mostra il nome dell'associazione di cui vogliamo cambiare le proprietà. La finestra è divisa in due parti, quella a sinistra relativa all'estremità dell'associazione collegata alla prima classe dell'associazione (la prima su cui si è fatto click mentre si creava l'associazione), quella a destra relativa all'estremità collegata con la seconda classe dell'associazione. In figura 4.16 nella parte sinistra vi sono i seguenti campi:

- *First Class* è un campo non modificabile che mostra il nome della classe che è connessa alla prima estremità dell'associazione: in questo caso è *Libro*.
- *Multiplicity* mostra la molteplicità relativa all'estremo dell'associazione cui è attaccata la classe *Libro* e indica come da specifica il numero di istanze della classe libro che possono partecipare all'associazione con una istanza della classe della parte opposta: in questo caso *Editore*.

- *Ordered* è una casella di spunta che indica se la molteplicità corrispondente si intende ordinata.
- *Aggregate*, analogamente, è un'altra casella di spunta associata alla prima estremità dell'associazione che se spuntata asserisce che la classe *Libro* è un *aggregato* rispetto alla classe *Editore* e l'associazione diviene un'aggregazione.
- *Composite* è come il caso precedente solo che se spuntata la classe *Libro* diviene una *classe composta* o *composto* rispetto a *Editore* e l'associazione diviene una *composizione*.
- *Navigable* indica se l'estremità corrispondente a *Libro* è navigabile rispetto a quella opposta, ovvero se le istanze di *Editore* possono riferirsi a tutte le istanze di *Libro* che partecipano alla relazione.
- *Role* è un campo ove si può dare il nome al ruolo della prima estremità (in questo caso *libro*) e si stabilisce la sua visibilità e se è derivata.

Nella parte destra ci sono gli stessi campi relativi all'estremità collegata ad *Editore*.

La molteplicità di un'estremità viene mostrata solo se diversa da 1. *Aggregato* e *composto* sono rappresentati con una losanga, rispettivamente bianca e nera, sull'estremità dell'associazione che è collegata alla classe che rappresenta l'intero. La freccia che indica che un'estremità è navigabile è indicata solo se l'estremità opposta non è navigabile. Il nome del ruolo è mostrato solo se specificato.

Si aggiunge ora un'altra associazione tra *Libro* ed *Autore* facendo passi simili ai precedenti e si ottiene il seguente diagramma.

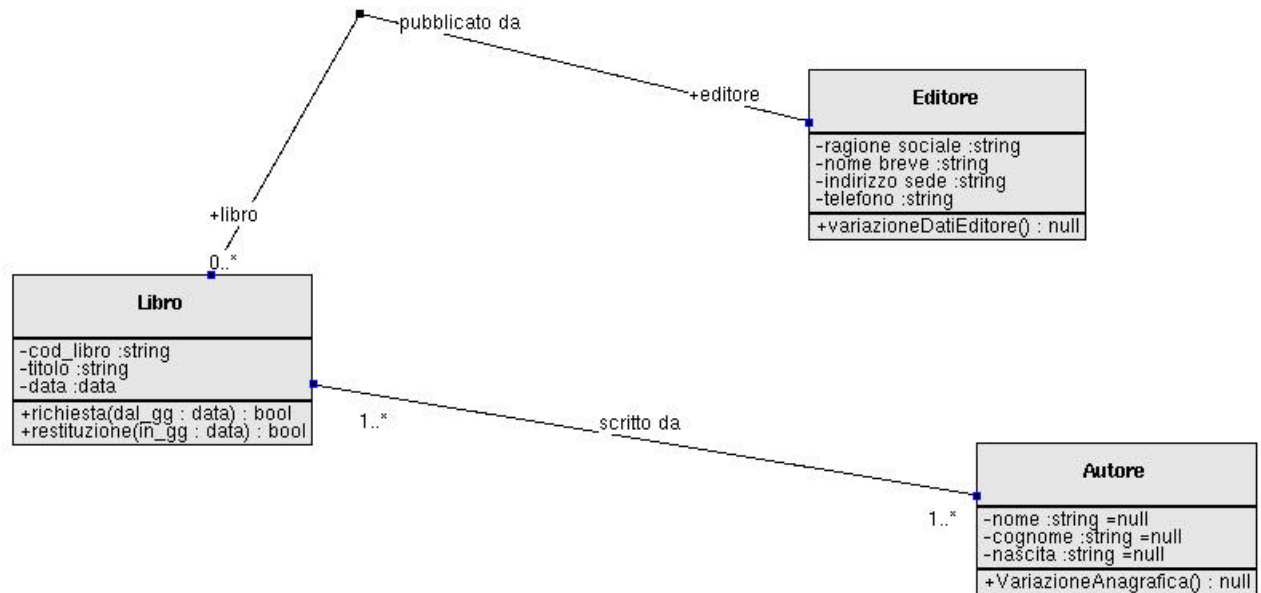



Figura 4.17 – Alcune classi legate da associazioni.

Se ora si vuole specificare che la classe *Libro prezioso* è una specializzazione di *Libro* si deve fare click sul tasto della barra dei bottoni: . Questo crea una nuova relazione di specializzazione. Una volta fatto click su questo tasto il cursore si trasforma, come nel caso dell'associazione, ad indicare che ora il programma aspetta di disegnare il cammino dalla classe figlia a quella padre. Quindi si opera come prima e si fa click su *Libro prezioso* e poi su *Libro*, ottenendo quanto illustrato in figura 4.18.

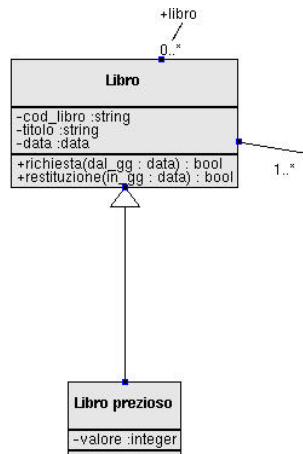


Figura 4.18 – Una specializzazione.

Per completare il modello si aggiungono la classe *Scaffale* e la classe *Biblioteca* e le relative associazioni, come si può notare nel seguente diagramma. La *Biblioteca* è una classe composita rispetto alla classe *Scaffale* e quest'ultima è un aggregato rispetto alla classe *Libro*.

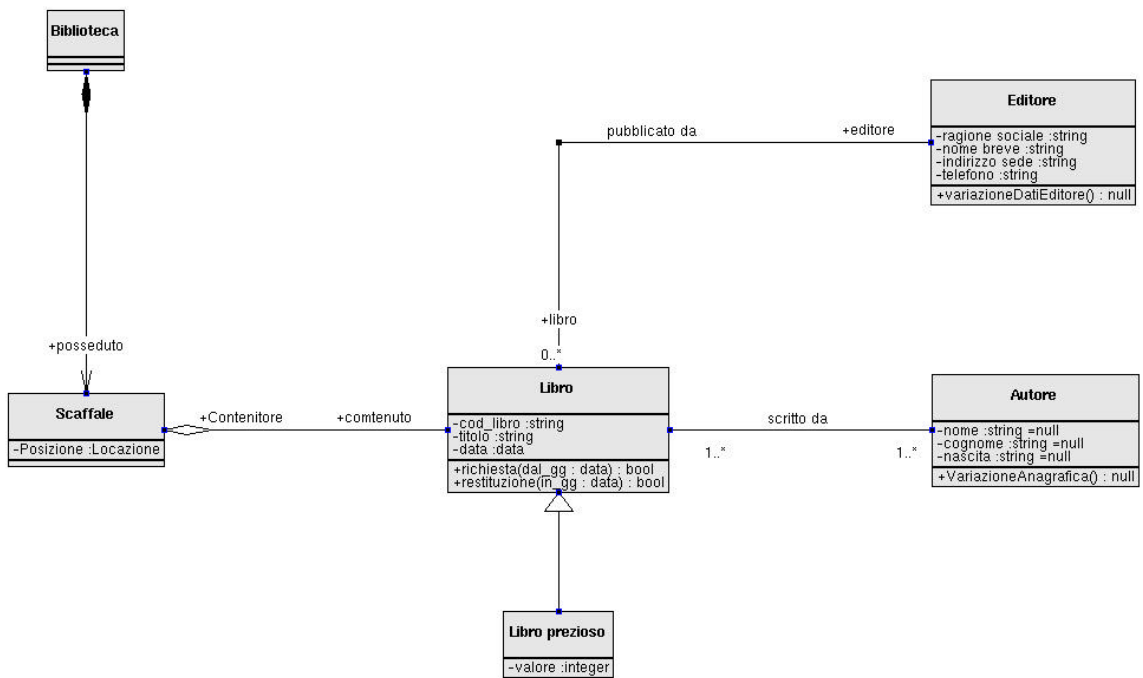







Figura 4.19 – Il diagramma delle classi di una biblioteca.

Gli altri bottoni della barra funzionano similmente: il bottone che crea una classe associazione è: . Se premuto il cursore si trasforma come nel caso dell'associazione e quindi si disegna il cammino tra le due classi che si vogliono legare con una classe associazione; quando si è finito compare il cammino tra le due classi con ancorato, tramite una linea tratteggiata, un rettangolo, come quello che rappresenta una classe, che può contenere gli attributi della classe associazione.

Il bottone che crea un'associazione n-aria è: . Se premuto crea un rombo che rappresenta l'associazione stessa e da cui partono tutti i cammini che portano alle classi coinvolte nell'associazione stessa. Per creare i cammini si fa click sul bottone associazione binaria e si unisce la prima classe che partecipa alla relazione con il rombo che rappresenta l'associazione n-aria e poi si ripetono questi passi per ogni altra classe che partecipa all'associazione n-aria.

Il bottone che crea una nuova dipendenza è: . Se premuto il cursore si trasforma a significare che si vuol disegnare il cammino dalla classe *client* a quella *supplier*. Se si vogliono legare con questa relazione insieme di classi va tracciato, come consuetudine, il cammino tra il primo *client* ed il primo *supplier* disegnando almeno un vertice di trascinamento lungo il cammino: questo primo vertice rappresenta il *NodeCross*, cioè il nodo che serve per aggiungere nuovi *client* e nuovi *supplier* alla relazione di dipendenza.

Per aggiungere un nuovo *client* si fa click su . Quindi si disegna il cammino tra il nuovo *client* ed il *NodeCross*.

Similmente per aggiungere un supplier si fa click su . Quindi si disegna il cammino tra il *NodeCross* ed il nuovo *supplier*.

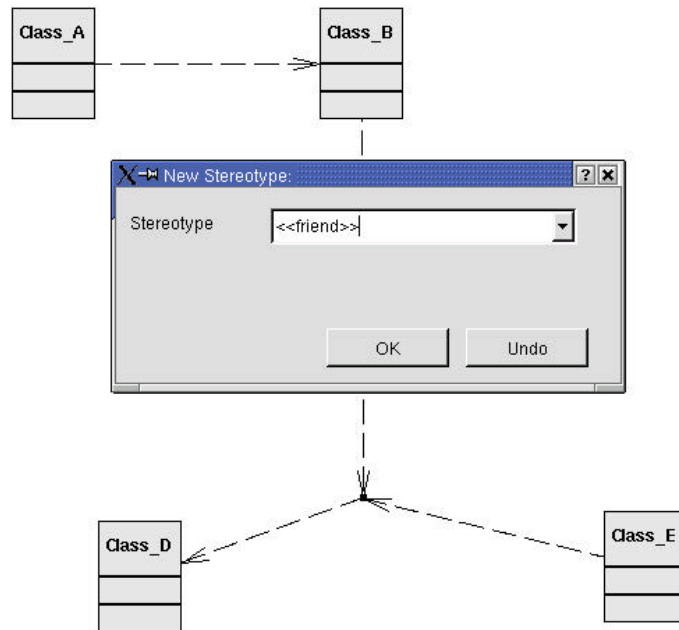


Figura 4.20 – La finestra che applica uno stereotipo.

Le dipendenze possono essere stereotipate premendo il tasto destro del mouse su un loro punto qualsiasi e scegliendo la voce *Stereotype* sul menù contestuale relativo. Appare quindi la finestra di dialogo mostrata in figura 4.20 con una combo box editabile che dà la possibilità di creare il nome dello stereotipo o di sceglierlo tra quelli già definiti precedentemente.

Conclusioni

Il lavoro propedeutico alla progettazione è servito a documentarsi in particolare sulla specifica dello UML [UMLS] ed in modo accessorio sulle altre specifiche di OMG (quali [MOFS] e [XMIS]) che sono in relazione con l'UML stesso. Inoltre si è presa coscienza dell'importanza dello sviluppo del software Open Source e delle vaste risorse che questo offre. Poi si è presa confidenza con la piattaforma offerta da Linux e con l'ambiente di sviluppo integrato di software in C ed in C++: K-Develop. Si è appreso l'uso delle librerie standard del C++ e delle librerie grafiche Qt.

Di conseguenza sono stati realizzati i seguenti progetti:

- la libreria in C++ CppUML per il supporto del metamodello UML: di questa è stato realizzato il codice sorgente del package *Foundation*;
- il tool per la modellazione UML e la realizzazione di un prototipo per la creazione di diagrammi di classe e del relativo modello tramite istanze della libreria CppUML.

Si deve completare l'implementazione di entrambi i progetti in base alle specifiche già presenti.