

Università degli Studi di Roma “La Sapienza”
Facoltà di Ingegneria – Corso di Laurea in Ingegneria Gestionale
Corso di Progettazione del Software
Proff. Toni Mancini e Monica Scannapieco

Progetto **PI.20070402**, passo **A.1**

versione del 10 aprile 2007

Si vuole progettare e realizzare *eBuy*, un sistema informatico per la gestione di aste on-line e di attività di commercio elettronico. In particolare, il sistema deve permettere, agli utenti registrati, di pubblicare annunci per la vendita di oggetti e di gestire aste al rialzo per la loro aggiudicazione. Il sistema deve anche consentire di mettere in vendita oggetti senza l'effettuazione di alcun'asta (formula “compralo subito”).

Si richiede di effettuare la fase di Analisi, producendo uno schema concettuale per l'applicazione i cui requisiti sono descritti in calce.

Requisiti

Il sistema deve permettere agli utenti registrati (di cui interessa il nome scelto e la data di registrazione) di pubblicare annunci per la vendita di *singoli* oggetti. Tali annunci sono chiamati *post*. Degli oggetti in vendita va specificata una descrizione, la categoria alla quale appartiene, il prezzo di vendita, i metodi di pagamento accettati (bonifico o carta di credito), e l'informazione sul fatto che l'oggetto sia nuovo oppure usato.

Gli annunci di vendita (*post*) sono di due tipologie distinte, potendo prevedere o meno un'asta al rialzo per la loro aggiudicazione. Per i *post* che prevedono un'asta, il venditore deve specificare il prezzo iniziale d'asta, l'ammontare dei singoli rialzi (ad es., 5 euro a rialzo) e l'istante di scadenza dell'asta. Al contrario, per i *post* che non prevedono un'asta (modalità di vendita “compralo subito”), al venditore è richiesto specificare esclusivamente il prezzo di vendita dell'oggetto. Il sistema deve consentire agli utenti (via Web) di pubblicare *post* per oggetti in vendita, con o senza asta.

Gli oggetti relativi a *post* che non prevedono asta vengono venduti al primo utente che procede con l'acquisto. I *post* che prevedono un'asta, invece, diventano oggetto di offerte di acquisto da parte di più utenti. Tali offerte vengono comunemente chiamate *bid*. Di ogni *bid* interessa l'istante in cui è stato proposto e l'utente offerente (chiamato *bidder*). Dato che i *post* oggetto d'asta specificano sia il prezzo iniziale che l'ammontare dei singoli rialzi, quando un *bidder* decide di proporre un *bid* per tale *post*, di fatto si propone di acquistare l'oggetto in questione

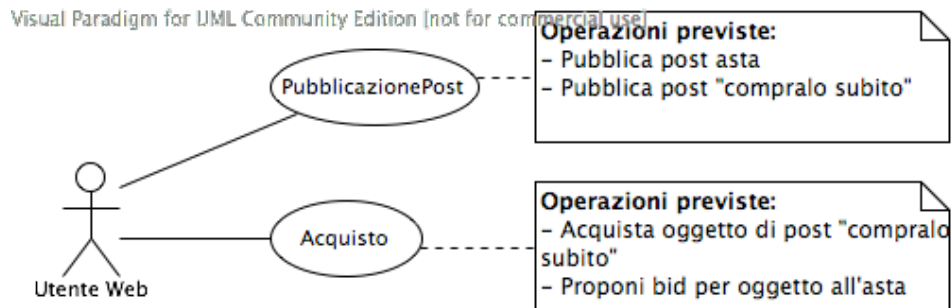
per un prezzo che è pari all'ultimo prezzo proposto fino a quel momento, aumentato dell'ammontare del rialzo (valore deciso a priori dal venditore). Ad esempio, se il prezzo del bid più recente è x euro e l'ammontare del rialzo è di r euro, il nuovo bidder si propone di acquistarlo per $x + r$ euro.

Il sistema deve consentire ad un utente (da Web) di proporre un nuovo bid per un oggetto in vendita tramite asta, oppure procedere all'acquisto di un oggetto messo in vendita con la modalità "compralo subito".

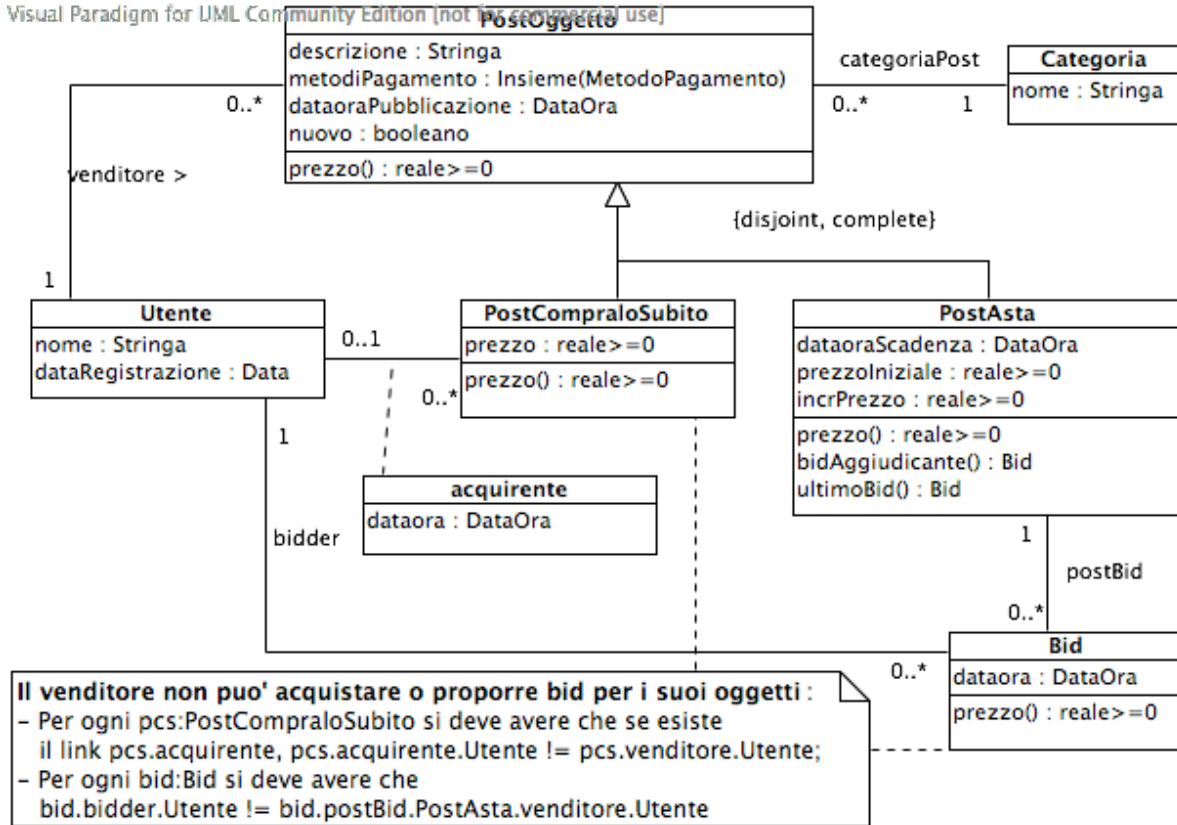
Le aste vengono automaticamente chiuse alla data/ora specificata dal venditore. A questo istante, l'ultimo utente che ha effettuato un bid si aggiudica l'oggetto in vendita, al prezzo del bid. Di ogni asta conclusa è di interesse conoscere il bid che si è aggiudicato l'oggetto in vendita (se esiste), con il prezzo relativo.

1 Fase di Analisi

1.1 Diagramma degli Use Case



1.2 Diagramma delle classi UML



1.3 Specifica dei tipi di dato

MetodoPagamento = {BONIFICO, CARTADICREDITO};

1.4 Specifica degli use case

```

SpecificaUseCase PubblicazionePost
    pubblicaPostAsta(
        utente:Utente,
        cat:Categoria,
        desc:Stringa,
        mpag:Insieme(MetodoPagamento),
        nuovo:booleano,
    )
    
```

```
    scad:DataOra,  
    prIniz:reale>=0,  
    incr:reale>=0): PostAsta  
pre: scad > adesso (con 'adesso' l'istanza del tipo DataOra che  
    denota l'istante corrente)  
post:  
result e' un nuovo oggetto di classe PostAsta con:  
- result.descrizione = desc  
- result.metodiPagamento = mpag  
- result.nuovo = nuovo  
- result.dataoraScadenza = scad  
- result.prezzoIniziale = prIniz  
- result.incrPrezzo = incr  
- result.dataoraPubblicazione = "adesso" (i.e., l'istanza del tipo DataOra  
    che denota l'istante corrente)
```

Inoltre, verranno creati i seguenti link:

- <result, utente>: venditore
- <result, cat>: categoriaPost.

```
pubblicaPostCompraloSubito(  
    utente:Utente,  
    cat:Categoria,  
    desc:Stringa,  
    mpag:Insieme(MetodoPagamento),  
    nuovo:booleano,  
    prezzo:reale>=0): PostCompraloSubito
```

```
pre: nessuna
```

```
post:
```

```
result e' un nuovo oggetto di classe PostCompraloSubito con:  
- result.descrizione = desc  
- result.metodiPagamento = mpag  
- result.nuovo = nuovo  
- result.prezzo = prezzo  
- result.dataoraPubblicazione = "adesso" (i.e., l'istanza del tipo DataOra  
    che denota l'istante corrente)
```

Inoltre, verranno creati i seguenti link:

- <result, utente>: venditore

- <result, cat>: categoriaPost.

FineSpecifica

SpecificaUseCase Acquisto

```
acquistoPostCompraloSubito(u:Privato, p: PostCompraloSubito)
  pre: p e' nello stato 'InVendita'
  post: viene creato il link <u, p>:acquirente con <u,p>.dataora = 'adesso'.
       Di conseguenza p passa nello stato 'Venduto' (cf. diagramma degli
       stati e transizioni della classe PostCompraloSubito)
```

```
proponiBid(u:Privato, p:PostAsta)
  pre: p e' nello stato 'AstaAperta'
  post: viene creato l'oggetto bid:Bid con bid.dataora = 'adesso'
       (istanza del tipo DataOra che denota l'istante corrente)
       e i seguenti link:
       - <bid, p>:postBid
       - <bid, u>:bidder
```

FineSpecifica

1.5 Specifica delle classi e diagrammi degli stati e transizioni

La classe PostOggetto

SpecificaClasse PostOggetto

```
prezzo(): reale >= 0
  pre: nessuna
  post: result dipende dalla classe piu' specifica di this
```

FineSpecifica

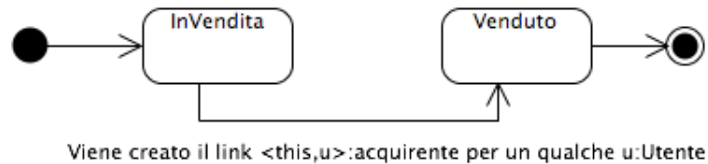
La classe PostCompraloSubito

SpecificaClasse PostCompraloSubito

```
prezzo(): reale >= 0
  pre: nessuna
  post: result = this.prezzo
```

FineSpecifica

Gli oggetti di questa classe evolvono secondo le regole imposte dal seguente diagramma degli stati e transizioni:



La classe PostAsta

Specificazione Classe PostAsta

prezzo(): reale ≥ 0

pre: nessuna

post: se $|this.postBid| = 0$ result = this.prezzoIniziale
altrimenti result = this.ultimoBid().prezzo()

ultimoBid() : Bid

pre: $|this.postBid| > 0$

post: result e' pari all'oggetto b:Bid tale che:

- $\langle b, this \rangle : postBid$

- per ogni altro $b' : Bid$ tale che $\langle b', this \rangle : postBid$, si ha che $b.dataora > b'.dataora$

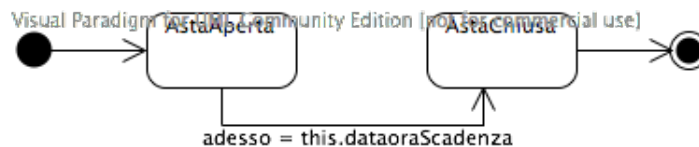
bidAggiudicante(): Bid

pre: this e' nello stato "AstaChiusa" e le prec. di this.ultimoBid() sono soddisfatte

post: result = this.ultimoBid()

Fine Specifica

Gli oggetti di questa classe evolvono secondo le regole imposte dal seguente diagramma degli stati e transizioni:



La classe Bid

SpecificaClasse Bid

prezzo(): reale ≥ 0

pre: nessuna

post: Detto $p = \text{this.postBid.PostAsta}$ (il post a cui il bid this si riferisce),

sia $B = \{ b:\text{Bid} \mid \langle b,p \rangle : p.\text{postBid} \text{ e } b.\text{dataora} < \text{this.dataora} \}$

l'insieme dei bid relativi all'asta p precedenti this .

$\text{result} = p.\text{prezzoIniziale} + (p.\text{incrPrezzo}) * |B|$

FineSpecifica