

Riflessioni su abcBid

Luca Cabibbo – 12 marzo 2007

Sommario

Scopo di questo documento è descrivere un buon possibile ragionamento da applicare alla soluzione del compito abcBid. La discussione è volutamente solo testuale, e non visuale. La discussione non è esaustiva.

Si suggerisce di leggere questo documento solo dopo: (1) aver studiato abbastanza e (2) aver provato a risolvere autonomamente il compito in questione.

I riferimenti, scritti tra parentesi quadre (ad esempio, [paragrafo 1.1]) sono riferimenti ad Applicare UML e i pattern, di C. Larman, Terza edizione.

Modellazione di dominio

Classi concettuali candidate

Quali le classi concettuali candidate per abcBid? Perché? [paragrafo 9.5]

- Offerta – “tentativo” di transazione commerciale (riguardano denaro, dunque si cominci da qui)
- Utente – ruolo di persona, correlato, tra l’altro, alle offerte, ma anche alla messa all’asta di oggetti
- Oggetto – prodotto correlato alle offerte – le offerte sono fatte per gli oggetti
- Il sistema abcBid – dove vengono registrati gli utenti, gli oggetti, le offerte
- Categoria – per organizzare gli oggetti, quindi possibile identificarle come “contenitori logici”
- Modalità d’asta – per descrivere il modo in cui andrà gestita l’asta per un oggetto

Altre osservazioni, circa l’uso delle generalizzazioni [paragrafi 31.2-31.9].

- E’ utile definire una gerarchia Venditore-Acquirente su Utente? E’ vero che si comportano in modo diverso, ma niente nei requisiti suggerisce che un utente non possa agire sia da Venditore che da Acquirente, nei confronti di oggetti diversi. Dunque è sbagliato farlo. Inoltre è francamente poco utile farlo. Per distinguere tra utente venditore e utente acquirente di un oggetto, è meglio usare delle associazioni [paragrafo 31.14].
- E’ utile definire delle sottoclassi Modalità AstaAlRialzo, ModalitàAstaConAggiudicazioneRapida e ModalitàAstaAlBuio, sottoclassi di ModalitàAsta? Sì, poiché le classi hanno attributi aggiuntivi di interesse, ed inoltre poiché il sistema si comporta diversamente, con modalità di interesse, in caso di modalità d’asta differenti [paragrafo 31.4] – in particolare, per determinare se un’offerta è ammissibile o meno (caso d’uso UC2, passo 5).

Altre osservazioni

- L’introduzione di una ulteriore classe Vendita è solo una complicazione.
- L’introduzione di una classe DescrizioneOggetto è solo una complicazione.
- L’introduzione di Pure Fabrication, come ad esempio delle Factory, è sbagliata nei modelli di dominio, che sono relativi ad un punto di vista concettuale.

Associazioni concettuali

Quali associazioni mostrare? [paragrafo 9.14] Quelle per cui è necessaria una certa memoria – tra operazioni di uno stesso caso d’uso, o tra casi d’usi diversi.

- Tra abcBid ed Utente – registra
- Tra abcBid ed Utente – per l’Utente corrente (ad esempio, ne è necessaria la memoria tra i passi 2 e 3 di UC1)
- Tra abcBid e Categoria
- Tra Utente ed Oggetto – vende
- Tra Oggetto e Categoria – classificato in
- Tra Offerta ed Utente – effettuata da
- Tra Offerta ed Oggetto – relativa a
- Tra Oggetto e ModalitàAsta – all’asta con modalità
- Tra abcBid ed Oggetto – per l’Oggetto corrente (in UC1, ne è necessaria la memoria tra i passi 3 e successivi. In UC2, ne è necessaria la memoria tra i passi 4 e 5)

Osservazioni

- L’associazione tra Oggetto e ModalitàAsta è bene che sia 1:1. Ovvero che ogni oggetto abbia la sua propria ModalitàAsta.
- E’ una complicazione associare un Utente alla ModalitàAsta anziché direttamente all’Oggetto.

- E' una complicazione associare un'Offerta alla ModalitàAsta anziché direttamente all'Oggetto.
- Riguardo all'associazione corrente con Oggetto. In effetti, nei casi d'uso UC1/2 è necessario ricordare l'oggetto corrente (in UC1, quello che si sta inserendo, in UC2, quello per cui si sta facendo l'offerta). Poco sopra è stato proposto di associare l'Oggetto corrente ad abcBid. In alternativa, è possibile associare l'Oggetto corrente all'Utente corrente. Quale delle due è preferibile? Ad occhio nessuna. In sede d'analisi (questa sede) è bene allora indicarle entrambe, indicandone una come /derivata [paragrafo 31.15]. Per le informazioni derivate (attributi ed associazioni), è bene mettere un commento per descrivere brevemente la rispettiva regola di derivazione. Fatto questo, nei contratti è possibile evitare di indicare la gestione di dati derivati, poiché si può assumere automaticamente derivata dalle regole di derivazione.
- Un'altra informazione significativa è, per ciascun Oggetto, l'Offerta attualmente migliore. Per ciò, potrebbe essere definita un'ulteriore associazione tra Oggetto ed Offerta (anche questa derivata).

Diagramma degli oggetti di dominio

Notare che il compito richiede di disegnarne uno. Dunque, l'unico diagramma deve contenere i due oggetti, le rispettive modalità d'asta, le rispettive categorie, e le rispettive offerte, nonché i tre utenti coinvolti, con tutti i collegamenti del caso.

Ciascun oggetto deve avere, tra l'altro, nome simbolico e tipo (classe). Ciascun collegamento deve essere etichettato con il nome della rispettiva associazione.

Diagrammi di sequenza di sistema

[capitolo 10] I nomi delle operazioni di sistema devono essere significativi, e riportare i parametri necessari. Le frecce tratteggiate di ritorno non indicano operazioni, e devono essere etichettate con le informazioni restituite, non con nomi di operazioni.

Caso d'uso UC1 – si noti che nel caso d'uso W2 va presa in considerazione solo la modalità d'asta al buio:

- Login(id utente, password)
- IniziaInserimentoOggetto(descrizione breve, descrizione dettagliata)
- AssociaOggettoACategoria(id categoria)
- AssociaModalitàAstaAlBuio(prezzo partenza, durata)

Durante la progettazione, bisogna considerare anche altre estensioni. Queste corrispondono ad altre due operazioni di sistema:

- AssociaModalitàAstaAlRialzo(prezzo partenza, durata)
- AssociaModalitàAstaConAggiudicazioneRapida(prezzo partenza, prezzo aggiudicazione rapida, durata)

Osservazioni

- Il testo del caso d'uso non suggerisce la presenza di un'interazione per confermare un oggetto. Pertanto è sbagliato mettere un'operazione di sistema ConfermaOggetto().

Caso d'uso UC2:

- Login(id utente, password)
- SelezionaCategoria(id categoria)
- SelezionaOggetto(id oggetto)
- EffettuaOfferta(importo)

Contratti delle operazioni di sistema

[capitolo 11] In particolare, devono essere solo dei tre tipi indicati nel [paragrafo 11.4].

Qui vengono discusse slno le post-condizioni. Non quello che va scritto nei contratti, ma il ragionamento.

Caso d'uso UC1:

- Login(id utente, password)
 - Al login, bisogna ricordare che quello è l'utente che da quel momento in poi usa il sistema. Va dunque formato un collegamento tra abcBid e Utente. Mediante che associazione? Non registra, ma quella dell'Utente corrente. Sulla base di che informazioni? Id utente. E la password? Il contratto può far riferimento ad una situazione di successo. Nessuna traccia nelle post-condizioni della password – a meno che uno voglia aggiungere un commento che non si confonda con le post-condizioni.
- IniziaInserimentoOggetto(descrizione breve, descrizione dettagliata)

- Che istanze vanno create? Un nuovo Oggetto. Che collegamenti vanno formati? Tra Oggetto e Utente corrente (questa è veramente importante). Bisogna anche ricordare che nelle operazioni successive si parla di questo Oggetto. Quindi anche un collegamento tra abcBid e Oggetto, corrente. Che attributi vanno cambiati? ...
- AssociaOggettoACategoria(id categoria)
 - Che istanze vanno create? Nessuna. Che collegamenti vanno formati? Tra Oggetto corrente e una Categoria. Quale Categoria? Sulla base di id categoria.
- AssociaModalitàAstaAlBuio(prezzo partenza, durata)
 - Che istanze vanno create? Una nuova ModalitàAstaAlBuio. Che collegamenti vanno formati? Tra ModalitàAstaAlBuio e Oggetto corrente. Che attributi vanno cambiati? ...

Osservazioni

- Al Login, va creato un nuovo Utente? Assolutamente no.
- Al Login, va bene dire come post-condizione “L’utente è stato autenticato”? Come post-condizione, assolutamente no. Come commento, se indicato chiaramente che è un commento, va bene.
- Ad AssociaOggettoACategoria, va creata una nuova Categoria? Assolutamente no.
- E’ grave non associare Oggetto e Utente? Oggetto e Categoria? Oggetto e Asta? Moltissimo, molto e molto.

Progettazione

Classi software

Con le scelte fatte, è abbastanza ragionevole immaginare che tutte le classi concettuali saranno rappresentate da opportune classi software nel DCD, con lo stesso nome. Probabilmente ci saranno anche altre classi software.

Associazioni – Visibilità

Con le scelte fatte, è abbastanza ragionevole immaginare che la maggior parte delle associazioni concettuali saranno rappresentate nel DCD. Forse non tutte. Forse anche qualcun’altra. In particolare, bisognerà fare delle scelte per le associazioni derivate.

Come noto, la visibilità è estremamente importante, soprattutto quella per attributo [capitolo 19].

Quindi è utile fare dei ragionamenti (da rivedere poi quando sia fa la progettazione con i diagrammi dinamici) sulle associazioni concettuali, per capire se è utile averle ed in che verso è utile averle (navigabili in un verso, nell’altro, oppure in tutti e due). Il criterio è: potrei volere l’associazione X navigabile da A e B se: (1) da qualche parte è possibile stabilire collegamenti in quel verso e (2) da qualche parte è utile navigare quei collegamenti in quel verso.

Riprendiamo le associazioni concettuali.

- Tra abcBid ed Utente – registra
 - Serve navigabile da abcBid a Utente? Sì, al Login di UC1/UC2. Possibile realizzare il collegamento in questo verso? Probabilmente nel caso d’uso di registrazione e in quello di avviamento.
 - Serve navigabile da Utente ad abcBid? Probabilmente no.
- Tra abcBid ed Utente – per l’Utente corrente (ad esempio, è necessaria la memoria tra i passi 2 e 3 di UC1)
 - Serve navigabile da abcBid a Utente? Sì, in IniziaInserimentoOggetto di UC1, nonché al passo 5 di UC2. Possibile realizzare il collegamento in questo verso? Sì, al Login di UC1/UC2.
 - Serve navigabile da Utente ad abcBid? No.
- Tra abcBid e Categoria
 - Serve navigabile da abcBid a Categoria? Sì, ad AssociaOggettoACategoria di UC1, nonché al passo 3 di UC2. Possibile realizzare il collegamento in questo verso? Nel caso d’uso di avviamento.
 - Serve navigabile da Categoria ad abcBid? No.
- Tra Utente ed Oggetto – vende
 - Serve navigabile da Utente ad Oggetto? Sì, in UC4. Possibile realizzare il collegamento in questo verso? Sì, in UC1.
 - Serve navigabile da Oggetto ad Utente? Sì, in UC3. Possibile realizzare il collegamento in questo verso? Sì, in UC1.
- Tra Offerta ed Utente – effettuata da

- Serve navigabile da Utente ad Offerta? Sì, in UC5. Possibile realizzare il collegamento in questo verso? Sì, in UC2.
- Serve navigabile da Offerta ad Utente? Sì, in UC3. Possibile realizzare il collegamento in questo verso? Sì, in UC2.
- Tra Offerta ed Oggetto – relativa a
 - Serve navigabile da Oggetto ad Offerta? Sì, in UC2/3/4/5. Possibile realizzare il collegamento in questo verso? Sì, in UC2.
 - Serve navigabile da Offerta ad Oggetto? Sì, in UC5. Possibile realizzare il collegamento in questo verso? Sì, in UC2.
 - Va gestita anche un'associazione tra Oggetto ed Offerta, per indicare l'Offerta migliore tra quelle finora pervenute? E' meglio fare questa scelta in sede di progettazione dei diagrammi di interazione.
- Tra Oggetto e ModalitàAsta – all'asta con modalità
 - Serve navigabile da Oggetto a ModalitàAsta? Sì, nel passo 5 di UC2. Possibile realizzare il collegamento in questo verso? Sì, in UC1.
 - Serve navigabile da ModalitàAsta ad Oggetto? Probabilmente no.
- Tra abcBid ed Oggetto – corrente (UC1, è necessaria la memoria tra i passi 3 e successivi. UC2, è necessaria la memoria tra i passi 4 e 5)
 - Serve navigabile da abcBid ad Oggetto? Sì, localmente ad entrambi i casi d'uso UC1/2. Possibile realizzare il collegamento in questo verso? Sì, localmente ad entrambi i casi d'uso UC1/2.
 - Serve navigabile da Oggetto ad abcBid? No
 - Alternativamente, si può considerare di gestire l'associazione tra Utente corrente ed Oggetto corrente. Una o l'altra è sufficiente. Dipende da scelte fatte sull'accoppiamento, che potranno essere fatte meglio durante la progettazione dei diagrammi di interazione.
- Tra Oggetto e Categoria?
 - Lasciato per esercizio. Quando mostro un oggetto, devo mostrare le sue categorie? Quando mostro una categoria, devono mostrare gli oggetti della categoria? Per rispondere devo sapere i requisiti, per sapere i requisiti devo leggere bene il testo del compito.

Progettazione per il caso d'uso UC1

Con gli obiettivi proposti di navigabilità da formare, è possibile procedere alla progettazione per le operazioni di sistema per il caso d'uso UC1. Per le responsabilità (in particolare, quali oggetti creare e quali associazioni formare), un buon punto di partenza sono i contratti (se fatti bene) e la discussione appena fatta sulle navigabilità.

- Controller
 - La prima scelta da fare è il Controller [paragrafo 17.13]. Un facade controller è la scelta più semplice, e probabilmente adeguata in questo caso. Chi scegliere? Probabilmente abcBid.
- Login(id utente, password)
 - Per IE, la password va probabilmente controllata dall'Utente il cui id è id utente. Ma prima bisogna trovare l'oggetto Utente il cui id è id utente – pattern “IDs to Objects” [pagina 475]. Per IE, può farlo solo chi conosce tutti gli Utenti. Dunque abcBid.
 - Dopo aver trovato l'Utente, abcBid lo ricorda come Utente corrente, formando un collegamento (indicarlo con una nota nel diagramma).
 - Per il controllo della password, bisogna chiedere all'Utente di verificare la sua password reale rispetto a quella passata come parametro dell'operazione di sistema. Il diagramma d'interazione probabilmente finisce lì. In pratica, se la password è diversa, l'Utente può segnalarlo restituendo un booleano, oppure, molto meglio, sollevando un'eccezione.
- IniziaInserimentoOggetto(descrizione breve, descrizione dettagliata)
 - Va creato un nuovo Oggetto. Chi lo crea? Bisogna applicare Creator [paragrafo 17.10]. Quali i suggerimenti di Creator? Utente, perché deve essere collegato all'Oggetto (con un collegamento bidirezionale, vedi la discussione sulle navigabilità). Oppure abcBid, perché dovrà usare strettamente l'Oggetto nel corso di questo caso d'uso. Quale è preferibile? In pratica, bisognerebbe provare a fare entrambi i progetti e misurare accoppiamento e coesione. Entrambe sono al momento accettabili, purché uno scriva un piccolo commento che dica “perché deve essere collegato all'Oggetto” se il creatore è Utente oppure “perché dovrà usare strettamente l'Oggetto nel corso di questo caso d'uso” se il creatore è abcBid.

- Che collegamenti vanno formati? Tra Oggetto e Utente corrente, bidirezionale. Pertanto, alla creazione dell'Oggetto bisogna passargli come parametro l'Utente corrente uc, ed in una nota indicare "this.venditore=uc". Inoltre, l'Utente corrente deve mettere l'Oggetto corrente in una lista di Oggetti. Facile se il creatore dell'Oggetto è l'Utente. Possibile se il creatore dell'Oggetto è abcBid. Ma glielo chiederà abcBid o l'Oggetto stesso?
- Che altri collegamenti vanno formati? Bisogna anche ricordare che l'Oggetto appena creato è l'Oggetto corrente. E' un'associazione diversa da quella formata poco sopra. E' il momento di capire se questa associazione va gestita dall'Utente corrente o da abcBid. Quale preferibile? Probabilmente è il caso di confermare la scelta fatta per il creatore di Oggetto. La cosa peggiore, in termini di accoppiamento, è forse fare ora una scelta diversa.
- AssociaOggettoACategoria(id categoria)
 - Che istanze vanno create? Nessuna. Che collegamenti vanno formati? Dalla Categoria il cui id è id categoria all'Oggetto corrente, inserendo l'Oggetto in una mappa della Categoria. Chi trova la Categoria – "IDs to Objects" [pagina 475]? abcBid.
 - E' utile che l'Oggetto gestisca una collezione delle sue Categorie? Come spiegato sopra, probabilmente è inutile, rispetto ai requisiti correnti. Quindi farlo è (probabilmente) sbagliato.

Per i passi 5/6, se non fosse per le estensioni, si tratterebbe di un'unica operazione di sistema, in cui le responsabilità del sistema sono indicate nel passo 6, ma l'unico evento da gestire è quello del passo 5. In realtà, diventano tre diverse operazioni di sistema a causa delle tre estensioni. Questo è analogo al passo 7 del caso d'uso *Elabora vendita* [paragrafo 6.8] che ha varie estensioni che rappresentano le varie modalità di pagare una vendita. Nel corso della progettazione, queste corrispondono ad operazioni di sistema diverse: per pagare in contanti [paragrafo 18.4], per pagare con carta di credito [paragrafo 32.1, figura 32.2], ...

Nell'esercizio W3, bisogna considerare solo la modalità d'asta al rialzo e quella con aggiudicazione rapida, quindi vanno mostrati due diversi diagrammi di interazione, relativi a due diverse operazioni di sistema. Consideriamo qui solo *AssociaModalitàAstaAIRialzo*, l'altra è simile.

- *AssociaModalitàAstaAIRialzo*(prezzo partenza, durata)
 - Che istanze vanno create? Una nuova *ModalitàAstaAIRialzo*. Chi la deve creare? Si noti che il creatore delle classi che estendono o implementano *ModalitàAsta* risulterà necessariamente accoppiato a tutte le sottoclassi di *ModalitàAsta*. Per *Low Coupling* [paragrafo 17.12], non è un accoppiamento molto bello, e quindi va probabilmente gestito con una *Pure Fabrication* [paragrafo 25.2], per non compromettere coesione/accoppiamento delle classi ispirate al dominio. Poiché è una responsabilità di creazione, va probabilmente usata una *Factory* [paragrafo 26.4], acceduta come *Singleton* [paragrafo 26.5]. Chiamiamo questo creatore *ModalitàAstaFactory*.
 - Ma chi chiede a *ModalitàAstaFactory* di creare la *ModalitàAsta* del tipo giusto? Questo oggetto dovrebbe probabilmente avere un diverso metodo per ciascuna modalità d'asta, il che richiede un accoppiamento con le varie modalità d'asta. Poiché il controller (abcBid) deve per forza avere un metodo per ciascuna operazione di sistema, ed in particolare uno per ciascuna diversa modalità d'asta, per ridurre l'accoppiamento complessivo è probabilmente bene che la richiesta della creazione della *ModalitàAsta* venga fatta da abcBid. Peggio (in termini di accoppiamento) farlo fare all'Oggetto o all'Utente. In questo caso, l'operazione richiesta sarà *creaModalitàAstaAIRialzo*, il che crea e restituisce un nuovo oggetto *ModalitàAstaAIRialzo*.
 - Che collegamenti vanno formati? Dall'Oggetto corrente alla *ModalitàAstaAIRialzo* appena creata. Quindi abcBid deve chiedere all'Oggetto corrente (direttamente, se ne ha visibilità, o indirettamente tramite l'Utente corrente) di eseguire un'operazione *setModalitàAsta*.

Progettazione per il caso d'uso UC2

Con gli obiettivi proposti di navigabilità da formare, è possibile procedere alla progettazione per le operazioni di sistema per il caso d'uso UC2. Si noti che la progettazione dovrà essere per le operazioni di sistema, che cambiano lo stato del sistema, non per le interrogazioni, che mostrano all'utente elenchi di categorie o oggetti. Possiamo evitare di farlo perché (e se) tutti i ragionamenti del caso sono stati fatti durante la discussione sulle navigabilità richieste. Evidentemente, questo è un buon momento per rivedere se le interrogazioni possono essere effettivamente soddisfatte con le navigabilità scelte.

Caso d'uso UC2:

- *Login*(id utente, password)
 - Come *Login* di UC1
- *SelezionaCategoria*(id categoria)

- Bisogna applicare “IDs to Objects” [pagina 475] per trovare la Categoria scelta dall’Utente. Chi il responsabile?
- Bisogna forse ricordarla anche come Categoria corrente (vedi operazione successiva). Quindi va probabilmente aggiunta un’associazione per la Categoria corrente (probabilmente da abcBid). Questa associazione non era stata finora prevista.
- SelezionaOggetto(id oggetto)
 - Bisogna applicare “IDs to Objects” [pagina 475] per trovare l’Oggetto scelto dall’Utente. Chi il responsabile? Due possibilità: abcBid (se conosce tutti gli oggetti) oppure la Categoria corrente.
 - Bisogna ricordarlo anche come Oggetto corrente (vedi operazione successiva). Chi il responsabile? Due possibilità... e tra queste non c’è Categoria.
- EffettuaOfferta(importo)
 - Questa è la parte più interessante. Se l’offerta è ammissibile (vedi passo 5 di UC2) allora va creata un’Offerta e collegata all’Oggetto corrente e all’Utente corrente (entrambi i collegamenti bidirezionali). Chi crea l’Offerta? Chi verifica se ammissibile? Chi forma i collegamenti?
 - Chi crea l’Offerta? Probabilmente l’Oggetto corrente o l’Utente corrente. La scelta peggiore, in termini di accoppiamento, è farla creare ad abcBid. Infatti, abcBid non ha bisogno di usare l’Offerta in operazioni successive, quindi non c’è nessuna giustificazione per farla creare a lui.
 - Quando creare l’Offerta? Prima o dopo di aver verificato che l’importo è ammissibile? In linea di principio, sembra lo stesso. In realtà, è meglio creare l’Offerta prima di verificare la sua ammissibilità, in modo tale che l’operazione che verifica l’ammissibilità prenda come parametro un oggetto, e non un valore. Vedi pattern “Passare come parametro l’oggetto” [pagina 475] – anche se lì il contesto è diverso, la sostanza è la stessa.
 - Chi può verificare se l’Offerta è ammissibile? La verifica dipende dalla modalità d’asta, quindi, per Polymorphism [paragrafo 25.1] sarà la ModalitàAsta. Poiché (per le scelte fatte finora) questa è visibile solo dall’Oggetto, bisognerà chiedere all’Oggetto corrente se l’Offerta è ammissibile, e l’Oggetto corrente chiederà alla sua ModalitàAsta. Ma allora è forse il caso che l’Offerta venga creata dall’Oggetto corrente, non dall’Utente corrente. Come verificare se l’Offerta è ammissibile? L’Oggetto corrente chiede (in modo polimorfo) alla sua ModalitàAsta di eseguire un’operazione verificaAmmissibilità, passandogli come parametro l’Offerta, ma anche un riferimento a se stesso (l’Oggetto corrente). La motivazione per quest’ultimo aspetto è la stessa per cui un oggetto contesto passa un riferimento a se stesso ad un suo oggetto strategy [paragrafo 26.7]. Vanno poi fatti diagrammi diversi per ModalitàAsta differenti.
 - Consideriamo il caso di ModalitàAstaAIRialzo. L’altra è simile (un po’ più complessa, in realtà, per il fatto che potrebbe andare chiusa l’asta). La ModalitàAstaAIRialzo chiede all’Oggetto corrente (ne ha visibilità per parametro) qual è il suo prezzo di partenza. Poi gli chiede l’importo dalla sua offerta migliore (questo è un buon motivo per avere anche l’associazione per l’Offerta migliore). Bisogna magari fare in modo che se l’Oggetto non ha ricevuto offerte allora risponde 0, o qualcosa del genere. Poi gli chiede se l’oggetto è ancora in vendita. La ModalitàAstaAIRialzo fa tutti i controlli del caso. Può restituire un booleano, oppure meglio l’oggetto Offerta oppure null, o forse meglio ancora un’eccezione se l’offerta non è ammissibile.
 - Se verificaAmmissibilità ha successo, allora vanno formati tutti gli altri collegamenti. Quello da Offerta a Oggetto e da Offerta a Utente è bene averli formati al momento della creazione dell’Offerta, passando queste informazioni come parametri del costruttore. Quelle in verso opposto è bene farle dopo la verifica di ammissibilità. L’Oggetto può farlo direttamente. Poi può chiedere all’Utente di aggiungere l’Offerta a quelle che ha fatto e che sono state accettate.
 - Notate che in questo modo abcBid non viene mai in contatto con oggetti Offerta. Se invece l’Offerta viene passata in giro, ne risulta un bel (anzi, un brutto) Blob. (Cerca con Google “antipattern blob”.)