Rational« XDE™ Linee guida per la struttura del modello per J2EE™

Documento di Rational Software
TP 154, 05/03

Tabella dei contenuti

1. Introduzione

Il presente documento fornisce raccomandazioni su come rappresentare e strutturare gli artefatti di modello di RUP« in Rational XDE™, Java Platform Edition. Naturalmente, decidere di modellare tali artefatti di RUP in XDE dipende dal progetto specifico. All'interno del presente documento, si notano i modelli a cui XDE fornisce supporto d'automazione e quelli a cui non lo fornisce poiché potrebbero influenzare una decisione.

Poiché tutti i modelli XDE esistono all'interno di progetti XDE, la sezione Struttura di progetto XDE fornisce raccomandazioni su quali progetti XDE e quali modelli al loro interno creare.

Sia RUP che XDE utilizzano il termine “modello” e l'associazione tra modelli RUP e XDE non è sempre nel rapporto uno ad uno. Nella sezione passaggio dal modello RUP al modello XDE, viene descritto tale passaggio.

La struttura di tutti gli artefatti del modello RUP nei propri file di modello XDE viene poi descritta nella sua sezione.

2. Ambito

Questo documento si focalizza sulla descrizione delle strutture di file di modello XDE raccomandate, non sul processo per lo sviluppo dei contenuti degli artefatti di RUP associati. Non descrive dettagliate euristiche per definire i progetti XDE che contengono i modelli XDE descritti. Per informazioni su come definire, sviluppare e modellare i contenuti degli artefatti di RUP, vedere RUP. Per ulteriori informazioni sui progetti, vedere la documentazione IDE.

Il presente documento non descrive un esempio completo ma utilizza quelli selezionati in base all'enfasi che pongono sui punti da coprire; tuttavia, tutti gli esempi sono coerenti l'un l'altro e sono presi dai modelli XDE attuali.

Questa versione del documento non affronta lo sviluppo di una libreria di tag.

Le strutture di progetto e di modello descritte nel presente documento sono solo raccomandazioni e possono essere sostituite da qualsiasi numero di strutture ugualmente valide.

3. Struttura del progetto XDE

L'attenzione è posta su come strutturare i modelli XDE. Tuttavia, poiché tutti i modelli XDE si trovano all'interno di progetti XDE, è importante fornire una breve introduzione alla struttura del progetto in cui si trovano le strutture del modello.

Per un'applicazione aziendale di J2EE sviluppata da più persone, si raccomanda la creazione dei seguenti progetti e modelli XDE.
Nota: Se le procedure guidate di “creazione progetto” di XDE vengono utilizzate, molti modelli saranno creati automaticamente insieme al progetto. Infatti, utilizzando WSS AD XDE, creando un progetto di modellazione dell'applicazione aziendale, gran parte di questa struttura di progetto multipla viene creata automaticamente insieme a molti modelli. XDE fornisce anche maschere di modello per saltare il contenuto dei modelli.

Progetto XDE	Descrizione	Modelli XDE "?<nome modello consigliato>" (<XDE tipo file: maschera modello>]
L'applicazione Progetto (Applicazione Modellazione di base progetto)	Il progetto d'applicazione rappresenta l'intero XDE. Contiene i file di modello XDE che descrivono l'applicazione nella sua totalità	- "?Modello di caso d'uso" (Rational XDE: Modello di caso d'uso) - "?Modello di analisi" (Rational XDE: Modello di analisi) - "?Modello di progettazione globale" (Rational XDE: Modello di progettazione) - "?Modello d'implementazione generale" (Rational XDE: Modello vuoto) - "?Modello di distribuzione EAR" - (Java: Modello di distribuzione EAR)
Modellazione dati logico (XDE Modellazione dati venditore)	Il progetto di modellazione dati contiene le risorse per le quali modellare i dati dell'applicazione, così come roundtrip engineer di un modello dati verso/da un database.	- "?Modello dati logico" (Dati: progetto il modello dati) - "?Modello dati fisico" (Dati: progetto del specifico file di modellazione dati fisico)) 1 - "?Modello di dominio" (Dati: file di modello di dominio specifico del venditore)
Progetti EJB (XDE EJB Modellazione)	I progetti EJB contengono le risorse necessarie per implementare EJB. Gli elementi contenuti sono impacchettati e distribuiti come modulo EJB (.EJB file JAR). È possibile definire separati progetti EJB per EJB individuali o serie di EJB (ogni progetto EJB può contenere al massimo un modello di codice Java). La raccomandazione è creare un progetto EJB per ogni EJB-JAR da produrre. Se vengono definiti progetti separati. allora il nome del progetto ne deve riflettere i contenuti.2	- "?Modello codice EJB" (Java: codice EJB Modello) - "?Modello di distribuzione EJB" (Java: progetto EJB Modello di distribuzione)
Progetti Web (Applicazione Web Modellazione JSP)	I progetti Web rappresentano le risorse Web del di XDE. Gli elementi contenuti sono impacchettati e distribuiti in un file di archivio Web (file WAR). progetti Web separati possono essere definiti per specifiche aree della logica di presentazione. La raccomandazione del Web è di creare un progetto Web per ogni WAR da produrre. Se vengono definiti progetti separati, allora il nome del progetto	- "?Modello di codice Java" (Java: Java 1.3/1.4 Modello di codice) - "?Modello di libreria tag JSP" (Web: Progetto Modello libreria tag)4 - "?Modello directory virtuale" (Web: Modello directory virtuale)5 - "?Modello di distribuzione Web" (Web: la Modello di distribuzione)

1 Rational XDE fornisce supporto al database fisico per più venditori database. Esiste un modello specifico di venditore per ogni venditore database supportato da XDE.

2 Utilizzando XDE per WSAD, se vengono creati progetti EJB aggiuntivi (Modellazione), la procedura guidata chiede un progetto d'applicazione per ospitare EAR. È necessario riutilizzare lo stesso progetto di applicazione (modellazione) di cui sopra.

3 Utilizzando XDE per WSAD, se vengono creati progetti Web aggiuntivi (Modellazione), la procedura guidata chiede un progetto d'applicazione per ospitare EAR. È necessario riutilizzare lo stesso progetto di applicazione (modellazione) di cui sopra.

4 Possono esistere più modelli di libreria tag per progetto. Infatti è necessario un modello separato per ogni file .tld. In ogni caso, questa versione del documento non affronta lo sviluppo della libreria tag.

5 Possono esserci più modelli di directory virtuale per il progetto Web di XDE.

	deve rifletterne i contenuti.[3]	

Un esempio di tale progetto ed organizzazione di modello è mostrato nella Figura 1 (notare i nomi di modello unici).

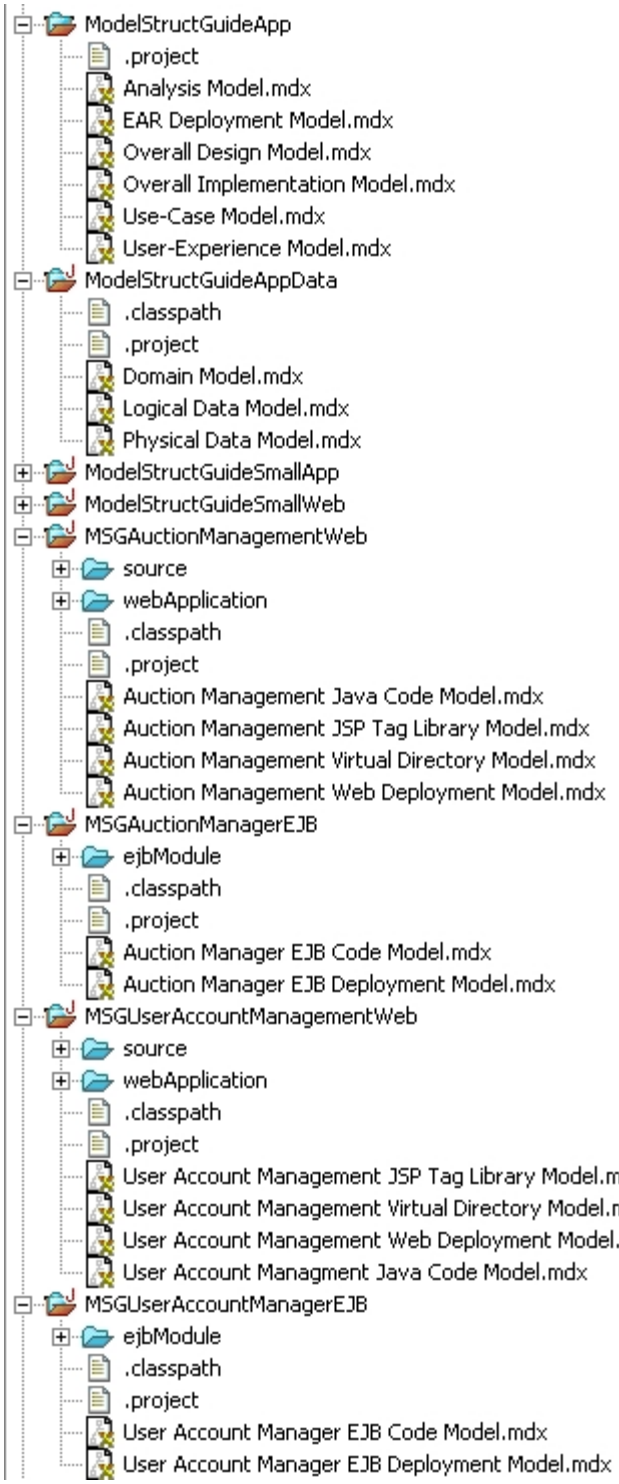


Figura 1: esempio di progetto XDE ed organizzazione di modello

In alternativa, se l'applicazione è davvero piccola e viene sviluppata da una sola persona, la struttura del progetto sopra descritto

può essere semplificata in due progetti, uno contenente l'applicazione generale ed elementi non Web e l'altro contenente gli elementi Web. Oltre a ridurre il numero di progetti, è possibile ridurre anche la quantità di modelli. Ad esempio nel caso di un progetto piccolo, con una sola persona, sono possibili le seguenti semplificazioni:

 Non viene mantenuto un modello di analisi separato. Analisi e progettazione vengono eseguiti entrambi nei modelli round-trip di XDE.

 Non vengono mantenuti un “modello di progettazione globale” ed un “modello d'implementazione generale” . Il progetto è abbastanza piccolo da poter ottenere una panoramica direttamente dai modelli round-trip di XDE. Vengono inoltre mantenute le realizzazioni di caso d'uso nel modello di codice EJB ed inclusi i riferimenti agli elementi nel modello di directory virtuale.

 Non viene mantenuto un modello dati logico. Uno schema dati fisico viene sviluppato direttamente nel “?Modello dati fisico”.

Questa “struttura di progetto piccolo” viene riepilogata nella tabella seguente.

Progetto XDE	Descrizione	Modelli XDE “?<nome modello raccomandato>” (<tipo file XDE: maschera di modello>]
L'applicazione Progetto (XDE Modellazione EJB Progetto)	Il progetto d'applicazione rappresenta gli aspetti non Web della applicazione. Contiene i modelli che descrivono l'applicazione nella sua totalità, il modello dati ed i modelli specifici di EJB.	- “?Modello di caso d'uso” (Rational XDE: modello di caso d'uso) - “?Modello dati fisico” (Data: vendor specific physical file di modello dati) - “?Modello codice EJB” (Java: modello codice EJB) - “?Modello di distribuzione EJB” (Java: modello di distribuzione EJB) - “?Modello di distribuzione EAR” (Java: modello di distribuzione EAR)
Progetto Web (Web XDE Modellazione Progetto)	I progetti Web rappresentano le risorse Web dell'applicazione. Gli elementi contenuti vengono impacchettati e distribuiti in un file di archivio Web (file WAR).	- “?Modello di codice Java” (Java: modello codice Java 1.3/1.4) - “?Modello di libreria tag JSP” (Web: libreria tag JSP Modello)6 - “?Modello directory virtuale” (Web: modello di directory virtuale)7 - “?Modello di distribuzione Web” (Web: modello di distribuzione Web)

6 Possono esistere più modelli di libreria tag per progetto. Infatti è necessario un modello separato per ogni file .tld. In ogni caso, questa versione del documento non affronta lo sviluppo della libreria tag.

7 Possono esserci più modelli di directory virtuale per il progetto Web di XDE.

Un esempio di piccolo progetto ed organizzazione del modello viene mostrato nella figura 2.

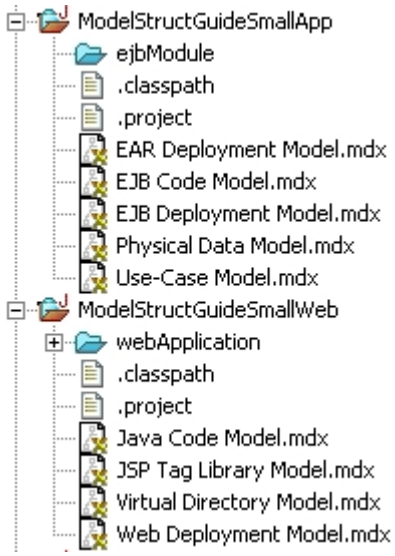


Figura 2: esempio di piccolo progetto XDE ed organizzazione di modello

L'attuale selezione della quantità di progetti e file di modello individuali risulta da una scelata strutturale e può variare per progetti differenti. Tuttavia, qualunque sia la quantità di progetti definiti, può esserci solo un file di modello di codice Java XDE per progetto. Per ulteriori informazioni sui progetti ed i file di modello XDE che possono contenere, vedere la documentazione di XDE.

È inoltre fortemente consigliato che i nomi del modello XDE siano unici lungo tutti i progetti XDE . Questo diviene estremamente importante quando si cerca di risolvere i riferimenti tra i modelli XDE. Per ulteriori informazioni sui riferimenti tramodelli e sulla loro risoluzione, vedere la documentazione XDE.

Per gli esempi nel presente documento, viene utilizzata la struttura di modello ed il progetto della Figura 1. Si noti che sono stati definiti più progetti EJB e Web. Per la logica dei progetti EJB e Web, vedere la sezione Sottosistemi d'implementazione.

4. Associazione modello RUP a modello XDE

Prima di descrivere come rappresentare gli artefatti del modello RUP in XDE, è importante indirizzarsi verso la confusione tra un "modello RUP" ed un "modello XDE" perché sono differenti e la mappatura dai modelli RUP ai modelli associati di XDE non è sempre in rapporto uno ad uno (ma quasi). Poiché il "modello" viene utiizzato sia in RUP che XDE, l'assunto iniziale è che debbano essere uguali. Tuttavia, i modelli in RUP separano le problematiche di processo (analisi contro progettazione contro implementazione, ecc..), mentre quelli in XDE separano quelle relative allo sviluppo (modelli di codici separati per la descrizione della struttura del pacchetto del linguaggio di programmazione contro una struttura di directory virtuale, modelli di codici separati per differenti linguaggi di programmazione ed ambienti di sviluppo, ecc..).
Per diminuire questa confusione, nel contesto del presente documento, il termine “modello” viene esplicitamente qualificato con “RUP” o “XDE”.

La seguente tabella riepiloga la mappatura dal modello RUP a quello XDE. I modelli XDE sono quelli introdotti nella sezione Struttura di progetto XDE. La struttura di ogni modello XDE viene descritta nelle sezioni successive del presente documento.

Modello RUP	<Progetto XDE>: < Nome modello XDE>
Modello del caso d'uso	Progetto di applicazione: modello di caso d'uso
Modello di analisi	Progetto di applicazione: modello di analisi
Modello di progettazione	Progetto di applicazione: modello di progettazione globale [Le classi di progettazione in ognuno dei file di modello round-trip di XDE (vedere sotto)]
Modellazione di dati	Modelli di dati XDE: - Progetto di modellazione dati: modello dati logico - Progetto di modellazione dati: modello dati fisico specifico del venditore - Progetto di modellazione dati: modello di dominio specifico del venditore
Progetto di applicazione del modello d'implementazione: generale	Modelli round-trip di XDE8 - Progetti EJB: modelli di codice - Progetti Web: modelli di codice Java - Progetti Web: modelli di libreria tag - Progetti Web: modelli di directory virtuale
Deployment Model (Modello di distribuzione)	Modelli di distribuzione XDE9 - Progetto di applicazione: modello di distribuzione EAR"10 - Progetti EJB: modelli di distribuzione EJB - Progetti Web: modelli di distribuzione Web

8 Per abbreviare, viene utilizzato il termine "modelli round-trip di XDE" lungo il presente documento per rappresentare tali modelli XDE.

9 Per abbreviare, viene utilizzato il termine "modelli di distribuzione XDE" lungo il presente documento per rappresentare tali modelli XDE.

10 Il modello di distribuzione EAR "incrocia" i modelli di distribuzione XDE individuali. Contiene diagrammi che descrivono i nodi della distribuzione e la loro connessione. Contiene anche diagrammi che mappano i file di archivio individuali, definiti nei modelli di distribuzione individuali, ai nodi di distribuzione.

5. Modello di caso d'uso

La struttura raccomandata del “modello di caso d'uso” viene mostrato nella Figura 3.

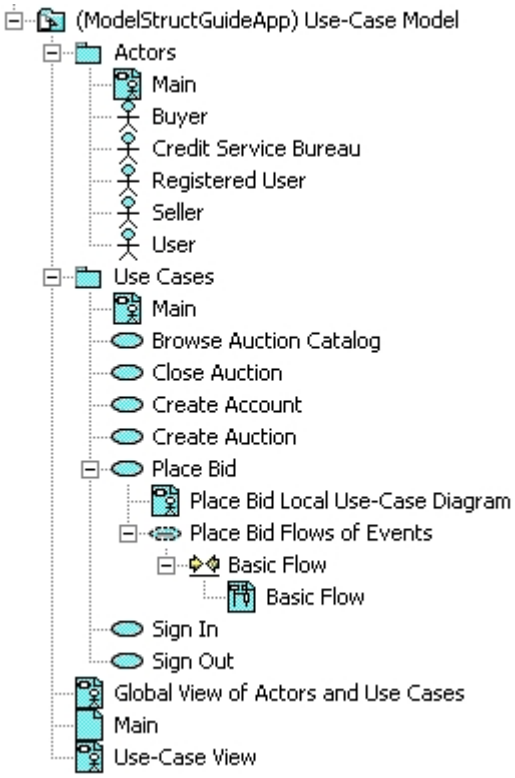


Figura 3: “modello di caso d'uso” Struttura

Il “modello di caso d'uso” viene diviso in due pacchetti: “Attori”e “Casi d'uso”.

Oltre ai diagrammi del modello di caso d'uso che contengono gli attori ed i casi d'uso, è possibile utilizzare diagrammi aggiuntivi per chiarire i diversi aspetti dei casi d'uso. È possibile includere i seguenti elementi di modello supplementari “sotto” l'elemento di modello del caso d'uso nel modello di caso d'uso, come mostrato nella Figura 3:

- Il diagramma “caso d'uso locale di esecuzione offerta” contiene il caso d'uso“esecuzione offerta” e gli attori che partecipano in quel caso d'uso.

- L'istanza di collaborazione “flussi di eventi di esecuzione offerta” contiene le istanze d'interazione che descrivono graficamente il flusso di eventi spiegato nella descrizione del caso d'uso (ad esempio, l'interazione tra gli attori ed il caso d'uso). Le istanze di collaborazione del caso d'uso non vanno confuse con le realizzazioni di caso d'uso, descritte sia nella sezione del modello di analisiche in quella delle realizzazioni del caso d'uso di progettazione , poiché le istanze di collaborazione nel “modello di caso d'uso” sono strettamente a “scatola nera” e non creano interazioni descritte di elementi all'interno dell'applicazione.

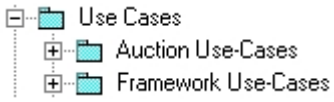
- L'attività grafica “flussi di eventi di esecuzione offerta” contiene i diagrammi di attività che descrivono graficamente il flusso di eventi spiegato nella descrizione del caso d'uso.

Nell'esempio nella figura 3, il diagramma di “vista globale di attori e casi d'uso” contiene tutti i casi d'uso e gli attori e le loro relazioni, diversamente dai diagrammi “principali” che contengono solo gli elementi nei pacchetti in cui tali diagrammi esistono. Nel caso di troppi attori e casi d'uso, le informazioni sul diagramma di “vista globale di attori e casi d'uso” possono essere espresse mediante diagrammi multipli.

Il diagramma di “vista di caso d'uso” rappresenta la vista di caso d'uso dell'architettura software. Per ulteriori informazioni sulle viste strutturali, vedere RUP.

Se si preferisce, possono essere creati ulteriori pacchetti all'interno di quelli degli “attori”e dei “casi d'uso” per organizzare ulteriormente gli elementi di modello contenuti come mostrato nella figura 4.

Figura 4: divisione aggiuntiva del pacchetto dei casi d'uso



6. Modello di analisi

Il modello di analisi è il luogo in cui risiedono le classi di analisi e le realizzazioni di caso d'usodi analisi.

Nota: decidere di mantenere o meno un modello di analisi ed uno di progettazione separati è una decisione specifica del progetto . Se un modello di analisiseparato viene creato ma non mantenuto, allora le classi di analisi verranno spostate nellapartizione appropriata del modello di progettazione [11] e rifinite. Un'altra opzione consiste nel creare le classi di analisi e le realizzazioni di caso d'usodi analisi nel modello di progettazionee poi farle evolvere da lì nella loro forma di progettazione. Vedere la sezione modello di progettazione per ulteriori informazioni su come il modello di progettazione viene rappresentato in XDE.

La struttura raccomandata per il modello di analisi viene mostrato nella figura 5.

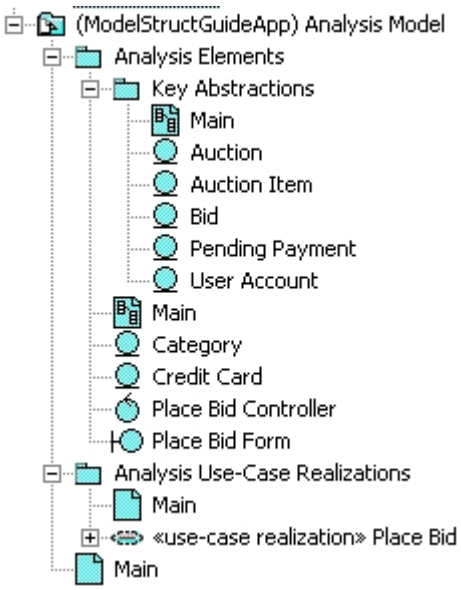


Figura 5: struttura del modello di analisi

Il pacchetto “elementi di analisi” contiene le classi di analisi. Le istanze delle classi di analisi appaiono nei diagrammi nel pacchetto delle "realizzazioni del caso d'uso di analisi".

Oltre alle classi di analisi, i pacchetti possono essere definiti in quello degli “elementi di analisi”per dividere ulteriormente le classi di analisicontenute (vedere il pacchetto delle “astrazioni chiave nella figura 5). Questa ulteriore divisione è facoltativa, specie se un modello di analisi separato non viene mantenuto. In tali casi, le classi di analisi possono essere considerate “transitorie” (ad esempio, esistono solo finché non evolvono in elementi di progettazione), perciò la loro organizzazione non viene considerata critica. Una possibile eccezione è l'astrazione chiave delle classi di analisi. Come mostrato nella figura 5, il pacchetto delle “astrazioni chiave” contiene le classi di analisi considerate per rappresentare le astrazioni chiave del sistema. Come precedentemente notato, il pacchetto è facoltativo. Un'alternativa consiste nel rappresentare le astrazioni chiave su un diagramma di classe nel pacchetto degli “elementi di analisi” . Tuttavia, creare un pacchetto separato fornisce una classificazione per categoria più esplicita delle classi di analisi come astrazioni chiave. Infatti, anche se un modello di analisi separato non viene mantenuto nella sua interezza, alcuni progetti possono scegliere di mantenere le classi di analisi dell'astrazione chiave.

11 Come si vedrà più tardi, la giusta “partizione del modello di progettazione” può essere solo un pacchetto in uno dei modelli round-trip XDE poiché la progettazione di elementi specifici della tecnologia viene eseguita nei modelli round-trip.

In tali casi risulta utile definire un pacchetto separato per contenere le classi di analisi che vengono mantenute. Nota: le astrazioni chiave appaiono anche nel diagramma di “vista logica: astrazioni chiave” nel “modello di progettazione globale”. Vedere la sezione modello di progettazioneper ulteriori informazioni.

Il pacchetto di “realizzazioni del caso d'uso di analisi” contiene il livello di analisi delle realizzazioni di caso d'uso, che descrivono come vengono eseguiti i casi d'uso in termini di classi di analisi nel pacchetto degli “elementi di analisi” . Ognuna delle realizzazioni di caso d'usodi analisi realizza un caso d'uso nel modello di caso d'uso, ha lo stesso nome di quel caso d'usoe la struttura come quella mostrata nella figura 6.

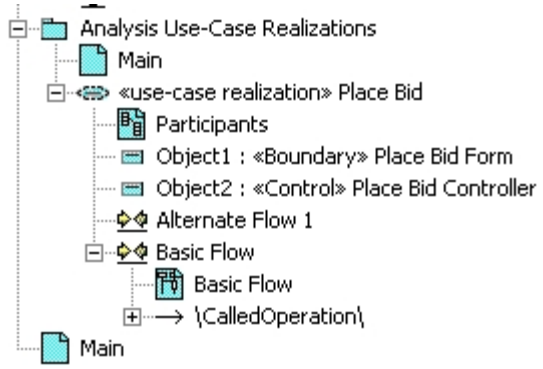


Figura 6: “realizzazione del caso d'uso di analisi” - struttura del pacchetto

Il diagramma dei “partecipanti” mostra le classi di analisi (dal pacchetto degli “elementi di analisi”) che partecipano alla realizzazione del caso d'uso (ossia, quelle classi di analisi le cui istanze appaiono nei diagrammi d'interazione) e nelle relazioni che supportano la collaborazione descritta nei diagrammi d'interazione.

Le istanze di interazione di "flusso" ("flusso di base" e "flusso alternato 1") contengono diagrammi di sequenza che descrivono i flussi di eventi del caso d'uso. Deve esserci un'istanza d'interazione per ogni flusso di eventi del caso d'uso significativo. I diagrammi di sequenza nelle istanze d'interazione descrivono il flusso tra le classi di analisi che partecipano durante l'esecuzione del caso d'usoassociato.

7. Modello di progettazione

Il modello di progettazione di RUP viene rappresentato da multipli modelli di XDE - il “modello di progettazione globale”dagli elementi di progettazione su cui è stato eseguito il round-trip che risiedono in modelli round-trip di XDE separati (questi sono dettagliati elementi di progettazione che partecipano al roundtrip engineering). In questo modo, l'automazione disponibile nei modelli round-trip individuali può essere potenziata. Ad esempio, i pattern EJB di XDE possono essere utilizzati per creare le classi che specificano EJB.

Il “modello di progettazione globale” descrive la progettazione dell'applicazione nel suo insieme e contiene elementi che attraversano multipli modelli round-trip di XDE. Contiene le partizioni logiche che ispirano l'organizzazione dei modelli round-trip individuali, così come le realizzazioni di caso d'uso che legano tutto insieme (le realizzazioni di caso d'uso descrivono la collaborazione tra gli elementi di progettazione a partire dai differenti modelli round-trip). Il “modello di progettazione globale” contiene diagrammi che fanno riferimento agli elementi di progettazione su cui è stato eseguito il round-trip. Per informazioni sui modelli round-trip di XDE individuali, vedere la sezione Modello d'implementazione.

Un'altra ossibilità consiste nel rappresentare il modello di progettazione ed il modello d'implementazionenello stesso modello di codice XDE. È possibile solo con un unico linguaggio d'implementazione di destinazione ed un team piccolo. Per un esempio di struttura di piccolo progetto, vedere la sezione Struttura progetto XDE.

Mantenere il “modello di progettazione globale” è facoltativo, ma è una buona idea per organizzare diagrammi, far emergere il livello di astrazione, ecc.., nonché fornire un posto per gli elementi di progettazione mentre si cerca di capire quale meccanismo d'implementazione applicare.

La struttura raccomandata del “modello di progettazione globale” viene mostrata nella Figura 7.

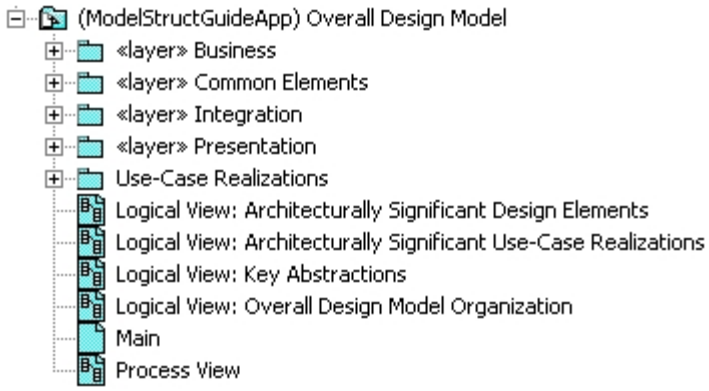


Figura 7: struttura del modello di progettazione globale

Il modello di progettazione globale contiene i seguenti pacchetti:

 I pacchetti di ½livello╗ contengono gli elementi di progettazione del sistema (o i diagrammi che ne fanno riferimento) (Le classi di progettazione, le interfacce ed i sottosistemi di progettazione). Questa struttura rappresenta una particolare strategia di divisione descritta nella sezione Livelli di progettazione.

 Il pacchetto “realizzazioni del caso d'uso” contiene le realizzazioni di caso d'uso al livello della progettazione. La struttura interna delle realizzazioni di caso d'uso viene discussa più in dettaglio nella sezione Realizzazioni di caso d'uso della progettazione

.

I diagrammi che rappresentano le viste strutturali includono “vista” nel nome del diagramma. Per ulteriori informazioni sulle viste strutturali, vedere RUP.

Il diagramma di “vista logica: astrazioni chiave” contiene le astrazioni chiave del sistema. Esistono diverse opzioni per mantenerle:

 Viene mantenuto un completo modello di analisi. In tal caso, il diagramma “vista logica: astrazioni chiave” contiene le classi di analisi dal modello di analisi che rappresentano le astrazioni chiave del sistema.

 Viene mantenuto un parziale modello di analisi, per la precisione solo le astrazioni chiavi. In tal caso, il diagramma “vista logica:
astrazioni chiave” contiene le classi di analisi dal modello di analisi che rappresentano le astrazioni chiave del sistema.

 Non viene mantenuta nessuna parte del modello di analisi. In tal caso, le classi di analisiche rappresentano le astrazioni chiave possono essere mantenute in un pacchetto nel modello di progettazione, chiamato “astrazioni chiave”

Per ulteriori informazioni sul modello di analisi, vedere la sezione Modello di analisi.

7.1 Livelli di progettazione

I pacchetti di ½livello╗ contengono gli elementi di progettazione del sistema (ad esempio, classi di progettazione, interfacce e sottosistemi di progettazione) che evolvono a partire dalle classi di analisi. I pacchetti di ½livello╗ possono contenere qualunque quantità di pacchettisecondari che dividono ulteriormente gli elementi di progettazione contenuti. Le realizzazioni di caso d'usodella progettazione (contenute nel pacchetto di “realizzazioni di caso d'uso” del “modello di progettazione” vengono discusse sotto l'intestazione nella sezione Realizzazioni di caso d'uso di progettazione) sono scritte nei termini degli elementi di progettazione contenuti in questi pacchetti.

Il modello di progettazione può seguire tutte le strategie di divisione. La strategia descritta in questa sezione è mostrata nella figura 8.

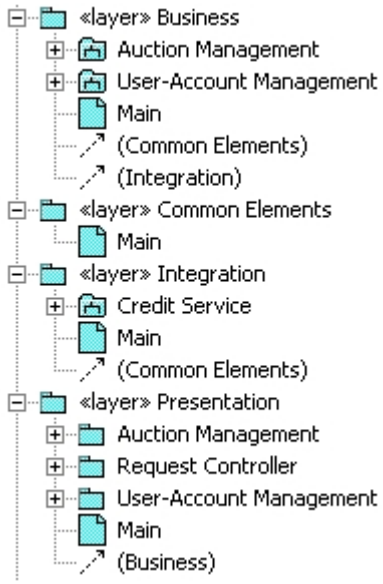


Figura 8: esempio di divisione del pacchetto di progettazione

In questo esempio, i pacchetti di primo livello sono considerati a loro volta livelli ed ognuno di essi ha una specifica responsailità. I pacchetti di secondo livello dividono ulteriormente gli elementi di pacchetto di livello per funzionalità dibusiness.

Il pacchetto del livello “presentazione” è responsabile della gestione delle interazioni con l'utente finale. In una applicazione J2EE, gli elementi di progettazione che possono risiedere nel pacchetto del livello “presentazione” includono pagine HTML, JSP (Java Server Pages) e servlet. È possibile dividere ulteriormente il pacchetto del livello “presentazione” in pacchetti secondari per raggruppare gli elementi che appartengono ad una serie correlata di casi d'uso; ad esempio, il pacchetto “gestione d'asta” nella figura 8.

Il pacchetto del livello “business” è responsabile dell'esecuzione di ogni processo business. Nella struttura di “modello di progettazione” presentata in questo documento, il pacchetto del livello “business” consiste in una serie di pacchetti di sottosistemi di progettazione, uno per ogni funzione business princiale (ad esempio, i pacchetti di sottosistemi “gestione d'asta” e “gestione account
utente”, nella figura 8). I pacchetti del sottosistema di progettazione vengono descritti in dettaglio sotto l'intestazione nella sezione Sottosistemi di progettazione.

Il pacchetto del livello “integrazione” è responsabile per l'accesso alle risorse di back-end, inclusi database e sistemi esterni. Nella struttura del modello di progettazionequi presentata, il pacchetto del livello “integrazione” comprende anche pacchetti del sottosistema di progettazione, uno per ogni sistema esterno (ad esempio, il pacchetto del sottosistema “servizio di credito” nella figura 8). I pacchetti del sottosistema di progettazione sono descritti più in dettaglio sotto l'intestazione nella sezione Sottosistemi di
progettazione.

Il pacchetto del livello “elementi comuni” contiene gli elementi condivisi tra i livelli.

Di nuovo, la struttura descritta in questa sezione può essere sostituita con una differente che riflette una diversa strategia di divisione.

7.2 Sottosistemi di progettazione

I sottosistemi di progettazione sono rappresentati da pacchetti del sottosistema nel “modello di progettazione globale”. Ogni pacchetto di sottosistema di progettazione deve avere la stessa struttura. Le specifiche di quella struttura variano in base al livello di dettaglio catturato per il sottosistema di progettazione.

Un esempio di una struttura di sottosistema di progettazione più formale e rigorosa viene mostrata nella figura 9.

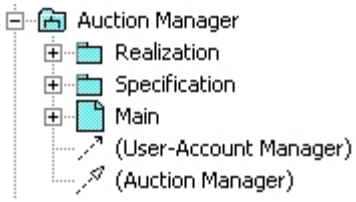


Figura 9: struttura del sottosistema di progettazione

La struttura del pacchetto del sottosistema di progettazione supporta la definizione dei pacchetti di "specifica" e di "realizzazione" all'interno del pacchetto del sottosistema di progettazione. Questa struttura è stata influenzata dal libro intitolato Componenti UML: Un semplice processo per la spiegazione del software basato sui componenti scritto da J. Cheesman e J. Daniels. Una struttura di pacchetto del sottosistema di progettazione semplificata che non contiene queste divisioni potrebbe essere utilizzata senza effetto sulle altre strutture del file di modello definite nel presente documento. Ognuno dei pacchetti di "specifica" e di "realizzazione" viene discusso nelle sezioni seguenti.

7.2.1 Specifica del sottosistema

Il pacchetto di "specifica" contiene una descrizione delle interfacce dei sottosistemi di progettazione.[12] Un esempio di una specifica di sottosistema viene mostrato in figura 10.

Figura 10: esempio di specifica del sottosistema di progettazione

7.2.2 Realizzazione del sottosistema

Il pacchetto "realizzazione" contiene una descrizione su come viene realizzata la specifica del sottosistema di progettazione. Un esempio di pacchetto "realizzazione" di un pacchetto di sottosistema di progettazione viene mostrato nella figura 11.

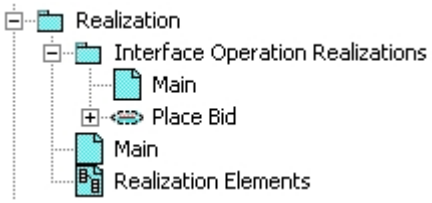


Figura 11: esempio di realizzazione del sottosistema di progettazione

Il diagramma degli "elementi di realizzazione" contiene riferimenti agli elementi di realizzazione del sottosistema. Gli elementi stessi possono risiedere nel pacchetto "realizzazione" o in un modello di codice XDE separato, in cui partecipano nel round-trip engineering. Per ulteriori informazioni, vedere la sezione Modelli round-trip di XDE

.

[12] In questo semplice esempio, è possibile che venga richiesto un pacchetto separato solo per l'interfaccia. Tuttavia, su un reale progetto il pacchetto va mantenuto perché può contenere riferimenti a documenti che descrivono il sottosistema e, in particolare, i limiti d'interfaccia come le condizioni precedenti e successive sulle operazioni.

Il pacchetto “realizzazioni dell'operazione d'interfaccia” contiene istanze di collaborazione che descrivono come gli elementi del sottosistema realizzano le importanti operazioni delle interfacce del sottosistema di progettazione(nel pacchetto “specifica”). Esiste una istanza di collaborazione per ogni operazione d'interfaccia del sottosistema importante.[13] Un esempio di un pacchetto “realizzazioni di operazione d'interfaccia” viene mostrato nella figura 12.

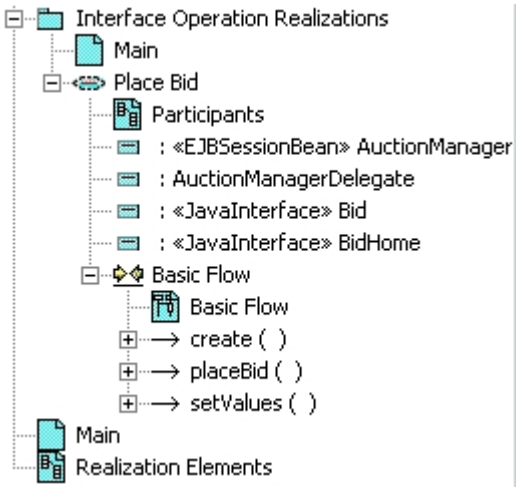


Figura 12: esempio di pacchetto di realizzazioni di operazione d'interfaccia

Così come per le realizzazioni di caso d'uso a livello di analisi (discusse nella sezione Modello di analisi) e le realizzazioni di caso d'usoa livello di progettazione (discusse più avanti nella sezione Realizzazioni di caso d'uso di progettazione), ogni realizzazione dell'operazione di interfaccia contiene un diagramma di classe con elementi del sottosistema che partecipano alla realizzazione (il diagramma “partecipanti” nella figura 12), così come diagrammi d'interazione che descrivono il tipo di collaborazione dei partecipanti all'esecuzione dell'operazione dell'interfaccia del sottosistema (il diagramma “flusso di base” nella figura 12).

7.3 Realizzazioni del caso d'uso di progettazione

Il pacchetto “realizzazioni del caso d'uso” contiene le realizzazioni di caso d'usoa livello di progettazione. Ognuna delle realizzazioni di caso d'uso è associata ad un caso d'uso nel modello di caso d'uso, ha lo stesso nome di quel caso d'usoe deve avere la struttura mostrata nella figura 16.

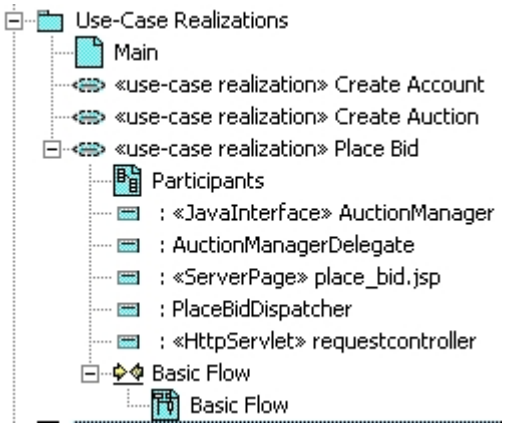


Figura 13: struttura di realizzazione del caso d'uso di progettazione

Il diagramma “partecipanti” delle realizzazioni di caso d'usomostra gli elementi di progettazione che partecipano nella realizzazione del caso d'uso (ossia, quegli elementi di progettazione le cui istanze appaiono nei diagrammi d'interazione della realizzazione di caso d'uso) e le relazioni che supportano le collaborazioni descritte nei diagrammi d'interazione.

[13] Non tutte le operazioni vanno definite a questo livello. Alcune operazioni più semplici possono non aver bisogno di istanze di collaborazione separate.

Il diagramma “flusso di base” è un esempio di diagramma d'interazione che descrive il flusso tra gli elementi della progettazione che partecipano durante l'esecuzione del caso d'usoassociato. Deve esserci un'istanza d'interazione per ogni flusso di eventi nel caso d'uso.

È importante notare che i diagrammi della realizzazione di caso d'usopossono (ed in genere lo fanno) contenere riferimenti agli elementi di progettazione che fisicamente risiedono in modelli round-trip di XDE separati. La realizzazione di caso d'usoè dove viene dimostrata la collaborazione tra elementi in modelli round-trip separati.

8. Modello dati

Il modello datidi RUP viene rappresentato da multipli file di modello XDE:

- Modello dati logico (facoltativo). Rappresenta il modello dati logico, che è una vista indipendente di applicazione della progettazione logica del database.
- Modello dati fisico. Rappresenta un modello dati fisico specifico del venditore di database. Contiene dettagliati elementi di modello per la definizione delle specifiche caratteristiche delle tabelle di database. Il file di modello XDE del “modello dati fisico” include anche gli artefatti d'implementazione specifici del database per implementare le tabelle in un database specifico del venditore.
- Modello di dominio (facoltativo). Rappresenta i tipi di dati specifici del venditore del database che possono essere utilizzati per definire tipi di dati consistenti lungo il “modello dati fisico”.

La separazione dei file di modello XDE fornisce ottima flessibilità per l'automazione supportata tra il modello di progettazione, il modello dati e il database fisico.

Ognuno di questi file di modello vengono descritti in dettaglio di seguito.

8.1 Modello dati logico (facoltativo)

Il modello dati logico può essere utilizzato in situazioni in cui il progetto ha l'esigenza di creare una rappresentazione di dati logici autonoma e relazioni fondamentali per la progettazione del database. La creazione di un Modello dati logico di XDE è facoltativo poiché il team di progettazione del database può trasformare invece classi di progettazione permanenti nel modello di progettazione in tabelle nel modello dati per creare la struttura iniziale di progettazione del database fisico direttamente nel modello dati fisico XDE (vedere la sezione Modello dati fisici riportata sotto).

Il modello dati logico XDE può essere diviso in pacchetti d'area di soggetto, se necessario. Tali pacchetti definiscono raggruppamenti logici di classi di entità. Il modello dati logico XDE può anche contenere un pacchetto di “elementi comuni” che contiene elementi di modello che attraversano le aree di soggetto.

I diagrammi con “vista” nel nome vengono usati per documentare la vista dati della struttura. Il diagramma “vista dati: organizzazione del modello dati logico totale” viene utilizzato per documentare l'alto livello di organizzazione dati del modello dati logico, come espresso nelle principali divisioni (ad esempio, pacchetti) del modello dati logico XDE. La “vista dati : elementi di dati logici chiave” viene usata per documentare gli elementi logici chiave del modello dati. Per ulteriori informazioni sulle viste della struttura, vedere RUP.

Un esempio di struttura consigliata per il modello dati logico viene mostrato nella figura 14.

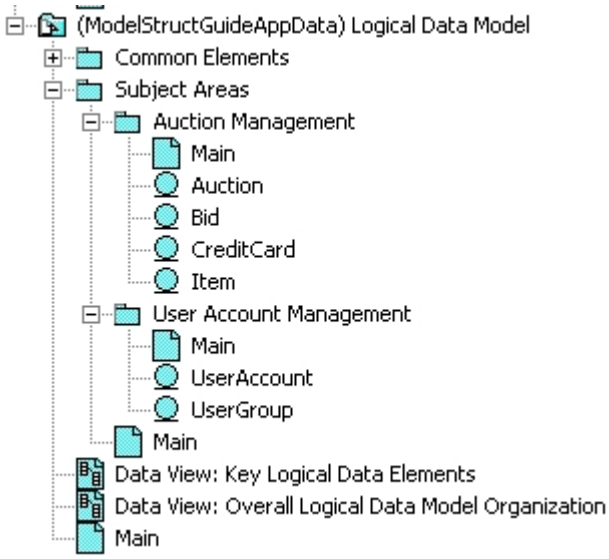


Figura 14: struttura modello dati logico

In questo esempio, esistono due pacchetti d'area di soggetto, “gestione d'asta” e “gestione account utente”. Ogni pacchetto d'area di soggetto contiene le classi di entità che insieme comprendono il modello dati logico. Non esiste una mappatura diretta alle strutture del pacchetto nel modello di progettazione anche se può esserci qualche somiglianza.

8.2 Modello dati fisico

Il modello dati fisico contiene la tabella di database dettagliata e progettazioni di procedura memorizzate che vengono usate per implementare il database attraverso i mezzi di forward engineering del modeler dei dati XDE. Il modello dati fisico è costituito inoltre dagli elementi del modello usati per definire la configurazione di memoria fisica del database. In generale, gli elementi del modello includono database e spazi tabella che comprendono il livello fisico delle tabelle di database sulla periferica di memorizzazione di destinazione.

Nella creazione del modello dati fisico, il progettista del database deve selezionare l'appropriato database di destinazione. I database supportati includono: DB2 MVS, DB2 UDB, Oracle, Sybase e SQL Server. XDE utilizza come valore predefinito il nome file di modello XDE per il database selezionato. Nell'esempio di “modello dati fisico” del presente documento, il nome file di modello XDE è stato aggiornato al “modello dati fisico”. Un progettista di database può scegliere di accettare il nome predefinito nella creazione del “modello dati fisico”.

Un esempio di struttura consigliata per il modello dati fisico viene mostrato nella figura 15.

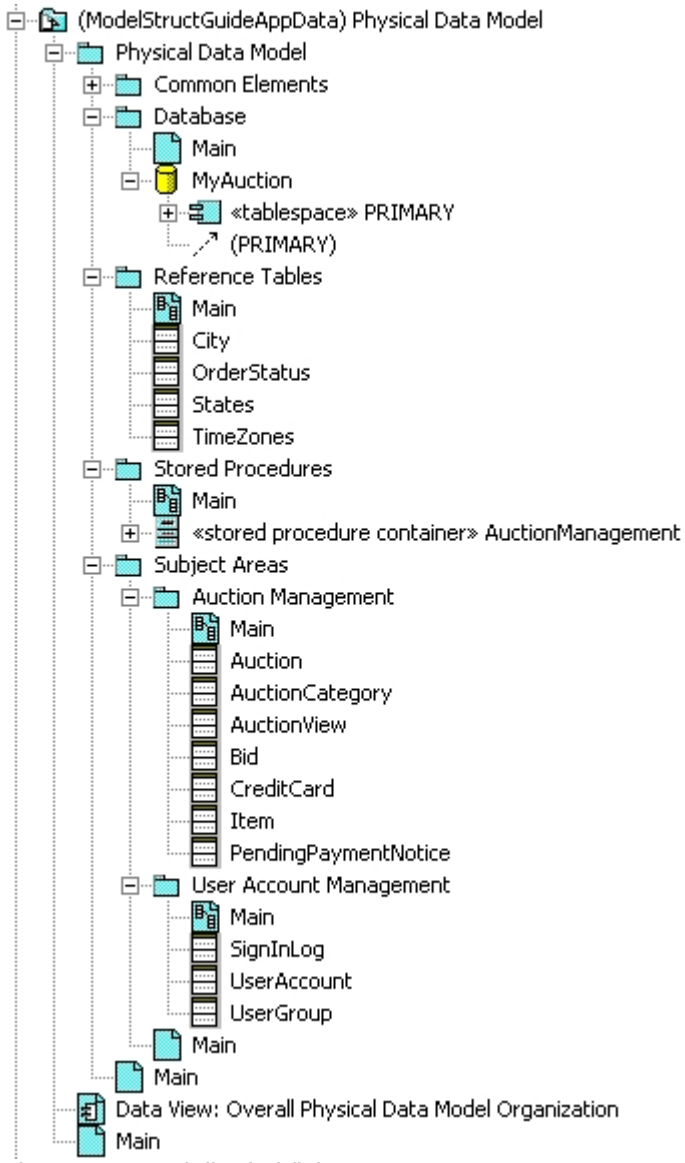


Figura 15: “modello dati fisico” Struttura

Il pacchetto “elementi comuni” contiene le viste e le tabelle database che attraversano le aree di soggetto.

Il pacchetto “database”contiene gli elementi di modello che definiscono la configurazione di memoria fisica del database. Contiene database e spazi tabella che comprendono il livello fisico delle tabelle di database sulla periferica di memorizzazione di destinazione. Vengono usati spazi tabelle per raggruppare logicamente le tabelle all'interno di un database. Per le linee guida sulla definizione degli spazi tabella, vedere RUP. Il pacchetto “database” può essere diviso in pacchetti di livello minore secondo necessità, in base alla complessità dell'applicazione.

Nell'esempio nella figura 15, il pacchetto “database”contiene un singolo database, MyAuction, lo spazio tabella ad esso associato, PRIMARY e le relazioni di realizzazione della tabella. Lo spazio tabella può essere denominato con ogni nome appropriato per un progetto di database. Per il database MyAuction, viene definito solo uno spazio tabella di nome PRIMARY.Se viene eseguito il forward engineering, vengono create le tabelle collegate al database attraverso la relazione con lo spazio tabella del database (sia in un database che in un DDL).

Il pacchetto “tabelle di riferimento” contiene le tabelle dei dati statistici che contengono informazioni di dati “costanti” necessari alla applicazione.

Il pacchetto “procedure memorizzate” contiene tutte le classi che rappresentano le procedure memorizzate del database (le classi di ½contenitore della procedura memorizzata ╗ e le associate operazioni ½di procedura memorizzata╗). Le procedure memorizzate che fanno riferimento ad una singola tabella possono essere impacchettate sia nel pacchetto “procedure memorizzate”che nel pacchetto “area di soggetto”con la tabella a cui fa riferimento la procedura, in base a cosa deve essere rappresentato, se una vista “incentrata sulla procedura memorizzata” oppure una “incentrata sulla tabella” [14].

Il pacchetto “aree di soggetto” contiene pacchetti che raggruppano logicamente serie correlate di viste e tabelle[15]. È consigliata la creazione delle viste nel pacchetto d'area del soggetto insieme alle tabelle. Questo consiglio è esclusivamente per motivi organizzativi. Può essere utile mantenere le viste in un'ara del soggetto in cui vengono usate, che le sistema nelle stesse aree del soggetto delle tabelle. Nell'esempio mostrato nella figura 15, esistono due pacchetti d'area del soggetto, “gestione d'asta” e “gestione account utente”. La quantità di pacchetti dell'area di soggetto dipende dalla complessità dell'applicazione. Tuttavia, in generale, tali pacchetti nel modello dati logico “ispirano” pacchetti d'area del soggetto nel modello dati fisico. Le aree del soggetto nel modello dati logico sono astrazioni delle rispettive aree nel modello dati fisico.

Le tabelle nei pacchetti d'area del soggetto contengono le colonne ed i trigger definiti per la tabella. Le tabelle vengono create attraverso

- Funzione di trasformazione da classe XDE a tabella.
- Eseguire il reverse engineering XDE su una funzione database esistente[16].
- Creazione manuale dal progettista database.

Durante l'esecuzione di reverse engineering su un database esistente , vengono creati pacchetti di schema nel modello dati fisico di XDE. I nomi di questi pacchetti si basano sul proprietario [17] del database su cui è stato eseguito il reverse engineering. Si consiglia lo spostamento delle tabelle su cui è stato eseguito il reverse engineering nei pacchetti d'area del soggetto all'interno del pacchetto “Aree di soggetto”e di eliminare i pacchetti di schema su cui è stato eseguito il reverse engineering. Lo spostamento delle tabelle nei pacchetti d'area del soggetto organizza in modo funzionale le tabelle per consentire al progettista database di aggiornare le tabelle secondo necessità.

I diagrammi con “vista” nel nome vengono usati per documentare la vista dati della struttura. Il diagramma “vista dati: organizzazione modello dati fisico generale” viene utilizzato per documentare l'alto livello di organizzazione dati del modello dati fisico, come espresso nelle principali divisioni (ad esempio, pacchetti) del modello dati fisico XDE. Per ulteriori informazioni sulle viste di struttura, vedere RUP.

8.3 Modello di dominio (facoltativo)

Il modello di dominio è un modello XDE facoltativo usato per memorizzare i tipi di dati definiti dall'utente per il database. I domini consentono ai progettisti database di riutilizzare le proprietà dell'elemento lungo la progettazione del database. Un dominio viene usato dal progettista database per documentare in modo costante le proprietà di una colonna dall'inizio alla fine del database. Il nome della colonna viene definito nella tabella; il dominio viene usato per definire il TypeExpression della colonna.

14 Una vista incentrata sulla tabella consente una migliore comprensione delle operazioni / progettazioni del database insieme. Una vista incentrata sulla procedura memorizzata semplifica l'individuazione e la modifica/mantenimento della procedura memorizzata.

15 Può sorgere qualche perplessità sull'uso di pacchetti d'area del soggetto nel modello dati fisico, poiché è necessaria ulteriore manutenzione per mantenere pacchetti d'area del soggetto di database fisico o logico. Tali aree si trovano qui per coerenza con il modello dati logico (se usato) e così nel caso in cui il modello dati fisico è “grande” e non c'è quello logico. In tal caso i pacchetti d'area possono essere usati per gestire le tabelle generate dalla trasformazione da classe a tabella.

16 In genere sul database viene eseguito una volta sola il reverse engineering e poi tutti i successivi aggiornamenti vengono sincronizzati mediante le funzioni Compare e Sync di XDE

17 All'interno di XDE, il proprietario del database viene acquisito come proprietà del componente di <<database>> . All'interno della proprietà di locazione, come parte della stringa di connessione, è presente un attributo di schema. Se viene eseguito il reverse engineering su un database, si tratta in genere del proprietario del database.

Un esempio di struttura consigliata per il modello di dominio[18] è mostrato nella figura 14.

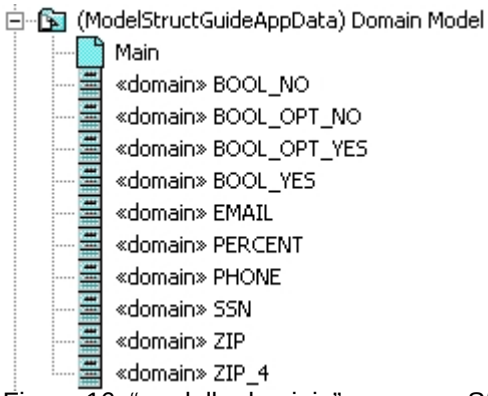


Figura 16: “modello dominio” Struttura

In questo esempio vengono mostrati i valori del dominio server SQL organizzati all'interno del pacchetto “dominio server SQL” . Nei casi in cui il progettista database definisce una grande quantità di domini, può essere necessario organizzarli mediante pacchetti sotto il pacchetto “dominio server SQL” .

9. Modello d'implementazione

Il modello d'implementazione di RUP viene rappresentato da più modelli XDE - il “modello d'implementazione globale” e più modelli round-trip XDE[19]. Questi ultimi tipi di modello consentono diverse riflessioni sull'implementazione da rappresentare più efficacemente, come la descrizione di una struttura a pacchetti Java rispetto ad una struttura di directory virtuale. Inoltre, l'automazione disponibile nei modelli round-trip individuali possono essere acquistati (ad esempio, XDE fornisce l'automazione per la creazione di elementi di modello con specifica tecnologia d'implementazione e la sincronizzazione di questi elementi di modello con il codice sorgente mediante un round-trip engineering).

Il “modello d'implementazione globale” descrive l'implementazione dell'applicazione nel suo complesso e contiene elementi che attraversano più modelli round-trip. Contiene diagrammi che fanno riferimento ad elementi negli elementi di di progettazione su cui è stato eseguito il round-trip e che in genere non “posseggono” alcun elemento di modello. Il “modello d'implementazione globale” non partecipa ad alcun round-trip engineering di XDE.

Mantenere il “modello d'implementazione globale” è facoltativo; tuttavia può rivelarsi utile per la descrizione della struttura globale dell'implementazione, incluso le unità d'integrazione (definite in RUP come sottosistemi d'implementazione) e per la documentazione della vista d'implementazione della struttura. I sottosistemi d'implementazionesono trattati in dettaglio nella sezione sottosistemi d'implementazione

La quantità di modelli round-trip di XDE dipende dall'organizzazione del modello e dal progetto XDE definito (per ulteriori informazioni, vedere la sezione struttura di progetto XDE). Tuttavia, una possibilità consiste nel definire progetti separati (e modelli round-trip) per ciascun sottosistema d'implementazione. Per informazioni su come rappresentare il sottosistema d'implementazionein XDE, vedere la sezione sottosistemi d'implementazione. In alternativa, come prima menzionato, con un singolo linguaggio d'implementazione di riferimento ed un team piccolo, è possibile scegliere l'uso di un singolo modello round-trip XDE per rappresentare sia il modello di progettazionedi RUP che il modello d'implementazione.

[18] All'interno di XDE sono supportati diversi database di venditori, incluso DB2, Oracle, Sybase e SQL Server. Per creare un modello dati XDE di dominio, il progettista seleziona l'appropriato database del venditore . XDE crea un elenco predefinito di domini per il venditore database selezionato.

[19] I modelli round-trip XDE sono ibridi. Sono usati per rappresentare sia i modelli di progettazione di RUP che la sua implementazione Modelli. Gli elementi di modello nei round-trip XDE che rappresentano le classi di progettazionedi RUP (le classi che mappano direttamente alle classi fisiche vengono considerate classi di progettazione) e i file d'implementazione fisica. Le strutture dei modelli round-trip di XDE rappresentano le strutture di directory fisiche.

Un esempio di “modello d'implementazione globale” è mostrato nella figura 17.

Figura 17: struttura di modello d'implementazione globale

Come mostrato nella Figura 17, il “modello d'implementazione globale” contiene solo diagrammi. I diagrammi che rappresentano viste strutturali includono “vista” nel nome del diagramma. Per informazioni dettagliate sulle viste strutturali, vedere RUP.

Il diagramma “vista d'implementazione: elementi d'implementazione distribuibile” fa riferimento ai file di archivio da distribuire ai nodi nei modelli di distribuzione di XDE. Gli stessi file di archivio sono fisicamente contenuti nei modelli di distribuzione. Per ulteriori informazioni, vedere la sezione modello di distribuzione.

Il diagramma “vista d'implementazione: sottosistemi d'implementazione” fa riferimento ai sottosistemi d'implementazione dell'applicazione. Possono essere tracciate relazioni di dipendenza tra i sottosistemi d'implementazione nel diagramma. Queste dipendenze rappresentano le importazioni del sottosistema d'implementazione ,che determinano l'ordine in cui i sottosistemi d'implementazionedevono essere integrati. Per informazioni su come rappresentare i sottosistemi di implementazione in XDE, vedere la sezione sottosistemi d'implementazione.

Il diagramma “vista d'implmentazione: struttura del modello d'implementazione” contiene riferimenti a tutti i modelli XDE usati per rappresentare il modello d'implementazione e le loro relazioni.
Nota: se vengono definiti progetti individuali per ognuno dei sottosistemi d'implementazione, allora il contenuto del diagramma “vista d'implementazione: struttura del modello d'implementazione” è ridondante rispetto alla “vista d'implementazione: sottosistemi d'implementazione” e può essere omesso. Per ulteriori informazioni sui sottosistemi d'implementazione, vedere la sezione sottosistemi d'implementazione.

9.1 Sottosistemi d'implementazione

I sottosistemi d'implementazione di RUP sono unità d'integrazione[20]. Dunque, si allineano bene con i moduli J2EE
(i sottosistemi d'implementazione possono essere impacchettati e distribuiti in moduli J2EE). Da qui un possibile modo per impostare
la struttura di progetto XDE consiste nel definirne uno per ogni archivio J2EE, da cui segue che
sottosistemi d'implementazione identificati possano essere utilizzati per guidare i progetti XDE da definire verso l'implementazione e
la progettazione dettagliata di supporto. In modo specifico, un sottosistema d'implementazione di RUP può essere rappresentato in XDE come suo
progetto. Questo è l'approccio utilizzato in queste linee guida in cui i sottosistemi d'implementazione vengono rappresentati mediante
progetti XDE.

I sottosistemi d'implementazione per gli esempi in queste linee guida sono:

- Un sottosistema d'implementazione per ogni sottosistema di progettazione nel “modello di progettazione globale”. Ad esempio: Responsabile account utente e responsabile asta.
- Un sottosistema d'implementazione per gli elementi di presentazione della gestione d'asta
- Un sottosistema d'implementazione per gli elementi di presentazione della gestione account utente

[20] Le linee guida per l'identificazione di sottosistemi d'implementazione non rientrano nell'ambito del presente documento. Per ulteriori informazioni, vedere RUP.

Progetti e modelli XDE associati sono mostrati nella figura 18.

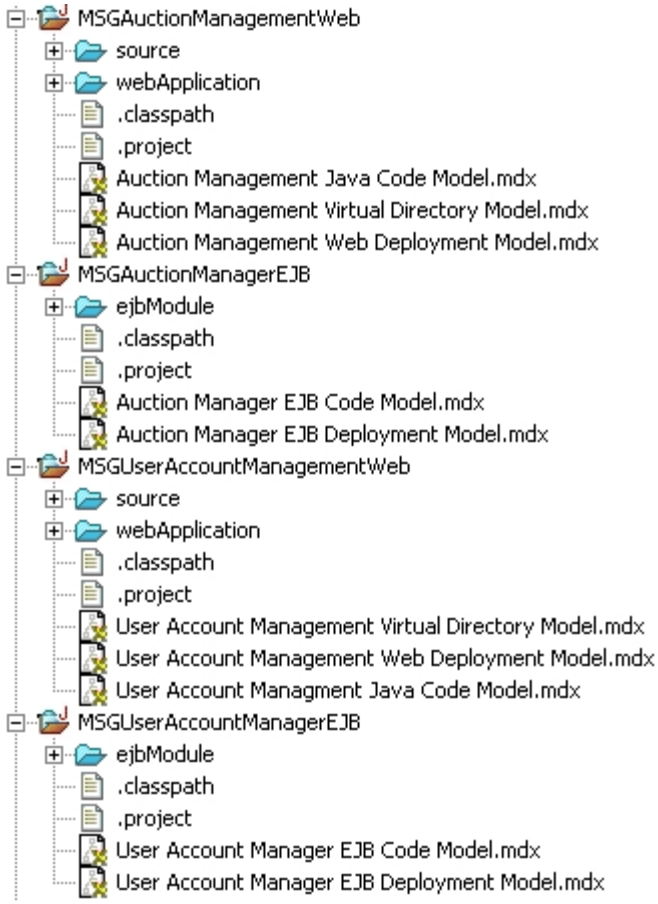


Figura 18: sottosistemi d'implementazione come progetti XDE

Se vengono definiti più progetti EJB e Web, è fortemente consigliato l'utilizzo di nomi del progetto e dei file di modello contenuti uguali. Questo rende più semplice l'interpretazione dei diagrammi nei modelli, così come risolvere riferimenti tra modelli. Nell'esempio mostrato nella Figura 18, il nome del sottosistema d'implementazione è stato utilizzato per il progetto e per tutti i modelli contenuti.

In alternativa, se il progetto è piccolo, è possibile rappresentare un sottosistema d'implementazionedi RUP in XDE come pacchetto in un modello XDE. La Figura 19 fornisce un esempio del caso in cui ogni sottosistema d'implementazione viene rappresentato mediante un pacchetto XDE (i pacchetti “auctionmanager” e “useraccountmanager”).

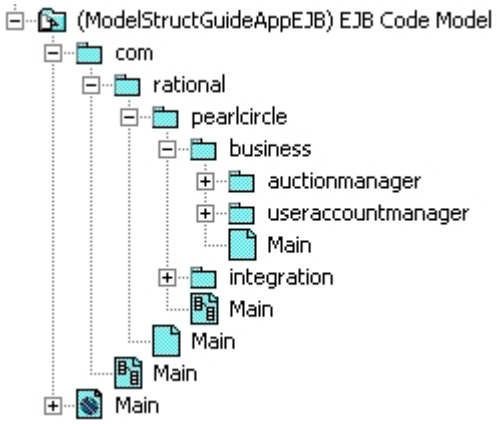


Figura 19: sottosistemi d'implementazione come pacchetti

9.2 Modelli round-trip di XDE

Questa sezione fornisce esempi per i seguenti modelli round-trip di XDE.

- (Progetto EJB) Modello codice EJB (vedere la sezione progetto EJB: modello codice EJB)
- (Progetto Web) Modello codice Java (vedere la sezione Progetto Web: modello codice Java)
- (Progetto Web) Modello directory virtuale (vedere la sezione Progetto Web: modello di directory virtuale)

In generale, strutturando i modelli round-trip XDE, il consiglio è di allineare la struttura il più vicino possibile alla struttura logica come descritto nel "modello di progettazione globale" (descritto nella sezione modello di progettazione), prendendo naturalmente in considerazione i limiti d'implementazione. Allineando la struttura del modello di progettazione e del modello d'implementazioneil più vicino possibile, la tracciabilità tra di loro diviene implicita e più
facile da mantenere. È importante finché la mappatura tra il modello di progettazione e quello di implementazionedeve essere mantenuta e gestita (come parte della manutenzione e
della gestione della struttura).

Le classi di progettazione che sono parti dei modelli round-trip XDE possono essere creati attraverso

- l'automazione XDE per la creazione di elementi dipendenti della tecnologia d'implementazione come EJB, servlet, etc., sia creandoli da zero che trasformando gli elementi indipendenti della tecnologia come le classi di analisi.
- il reverse engineering XDE di una implementazione esistente
- Creazione manuale.

9.2.1 Progetto EJB: modello codice EJB

Un modello di codice EJB contiene le risorse di Java necessarie all'implementazione di EJB (ad esempio, classi bean d'implementazione, classi d'interfaccia home, ecc..).

La proprietà principale di origine del modello di codice EJB deve essere impostata sulla directory in cui il codice sorgente viene posto. Può essere impostata, ad esempio, nella sotto-directory "ejbModule" del progetto EJB[21].

[21] Nel caso di un progetto che utilizza la procedura guidata per creare XDE, la sotto-directory del progetto viene creata automaticamente e la proprietà principale di origine del modello di codice viene impostata automaticamente.

Un esempio di modello di codice EJB viene mostrato nella figura 20.

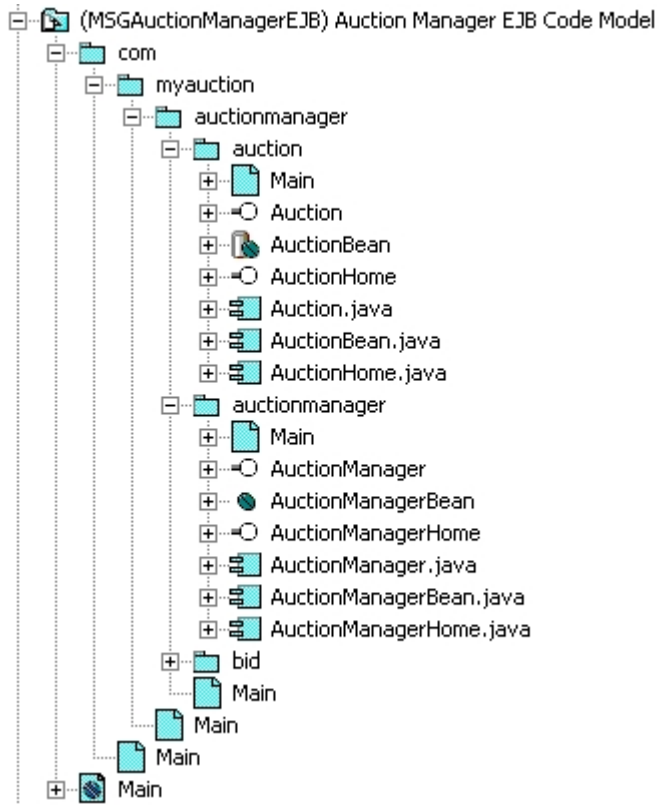


Figura 20: esempio di modello di codice EJB

Il modello di codice EJB mostrato nella Figura 20 contiene le risorse Java che implementano il sottosistema di progettazionedel responsabile asta. È l'omologo dell'implementazione al pacchetto di sottosistema di progettazione"gestione d'asta" del pacchetto di livello "business" nel "modello di progettazione globale" discussa nella sezione livelli di progettazione.

Come illustrato nella Figura 20, la struttura del modello di codice EJB segue la convenzione di utilizzo del nome di dominio come nome di pacchetto Java iniziale. Il nome di dominio per l'applicazione di esempio è "www.myauction.com". Perciò, i pacchetti contenenti gli elementi d'implementazione vengono posti all'interno del pacchetto "myauction" nel pacchetto "com" . Come risultato, tutti gli elementi Java all'interno del pacchetto "myauction" avranno un nome pienamente qualificato prefissato con "com.myauction". Ad esempio, il nome pienamente qualificato del pacchetto "asta" è "com.myauction.business.auctionmanager.auction". La convenzione di utilizzo del nome di dominio come nome di pacchetto Java iniziale garantisce che i nomi della classe di quest'ultimo siano unici, anche se viene incorporata una libreria di classi Java sviluppata da terze parti.

All'interno del pacchetto "myauction" , la struttura riflette quella del "modello di progettazione globale" (discussa nella sezione modello di progettazione). È presente un pacchetto per ogni livello di progettazione che contiene il sottosistema di progettazione del responsabile d'asta (pacchetto "business") e poi un altro pacchetto che rappresenta il sottosistema di progettazione del responsabile d'asta (pacchetto "auctionmanager"). Altri pacchetti sono stati definiti nel modello di codice EJB per raccogliere elementi di modello correlati (come ad esempio i pacchetti "auctionmanager", "auction" e "bid").
Nota: poiché i linguaggio di programmazione Java non consente spazi nei nomi di pacchetto, un nome potrebbe non essere identico a quello del pacchetto associato "modello di progettazione globale" .

Come mostrato nella Figura 20, un modello di codice EJB non solo contiene la rappresentazione visiva dei file di codice sorgente (gli elementi .java), ma anche le classi che specificano tali elementi. Queste classi rappresentano le classi di progettazione di RUP che si sono evolute fino al punto in cui è possibile implementarle e, nel caso di XDE, assegnare loro un round-trip engineering. Nota: XDE fornisce pattern che creano automaticamente tutte le classi usate per specificare un EJB.

9.2.2 Progetto Web: modello codice Java

Un modello di codice Java del progetto Web contiene le risorse Web di Java (come JavaBean, servlet, classi di assistenza, ecc.).

La proprietà principale di origine del modello di codice Java del progetto web deve essere impostato alla directory in cui il codice sorgente viene posto. Può ad esempio essere impostata alla sotto-directory “Java Source” del progetto web[22].

Un esempio di modello di codice Java del progetto web” viene mostrato nella Figura 21.

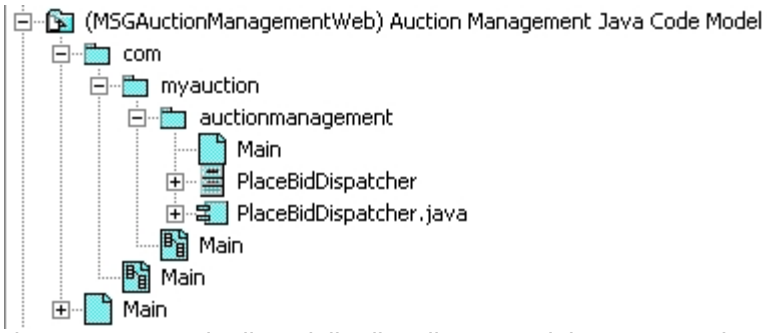


Figura 21: esempio di modello di codice Java del progetto web

Il modello mostrato nella figura 21 contiene le risorse Java che implementano gli elementi di progettazione di presentazione della gestione d'asta. È l'omologo dell'implementazione Java al pacchetto “gestione d'asta” del pacchetto del livello “presentazione” nel “modello di progettazione globale” discussa nella sezione livelli di progettazione).

Come illustrato nella Figura 21, la struttura del modello di codice Java del progetto web segue la convenzione di utilizzo del nome di dominio come nome di pacchetto Java iniziale. Il nome di dominio per l'applicazione di esempio è “www.myauction.com”. Perciò, i pacchetti contenenti gli elementi d'implementazione vengono posti all'interno del pacchetto “myauction” nel pacchetto “com” . Come risultato, tutti gli elementi Java all'interno del pacchetto “myauction” avranno un nome pienamente qualificato prefissato con “com.myauction”. Ad esempio, il nome pienamente qualificato del pacchetto “auctionmanagement” è “com.myauction.presentation.auctionmanagement”. La convenzione di utilizzo del nome di dominio come nome di pacchetto Java iniziale garantisce che i nomi della classe di quest'ultimo siano unici, anche se viene incorporata una libreria di classi Java sviluppata da terze parti.

All'interno del pacchetto “myauction” , la struttura riflette quella del “modello di progettazione globale” (discussa nella sezione modello di progettazione). È presente un pacchetto per ogni livello di progettazione che contiene gli elementi di presentazione della gestione d'asta (il pacchetto “presentazione”). E poi è presente un pacchetto che rappresenta gli elementi della presentazione della gestione d'asta (il pacchetto “auctionmanagement”).
Nota: poiché il linguaggio di programmazione Java non consente spazi nei nomi di pacchetto, un nome potrebbe non essere identico a quello del pacchetto associato “modello di progettazione globale” .

Come mostrato nella Figura 21, un modello di codice Java del progetto web non solo contiene la rappresentazione visiva dei file di codice sorgente (gli elementi .java), ma anche le classi che specificano tali elementi. Queste classi rappresentano le classi di progettazione di RUP che si sono evolute fino al punto in cui è possibile implementarle e, nel caso di XDE, assegnare loro un round-trip engineering.

9.2.3 Progetto web: modello di directory virtuale

Un modello di directory virtuale contiene risorse web di codice sorgente diverso da Java (ad esempio JSP e pagine HTML). Possono esserci più modelli di directory virtuale per il progetto Web di XDE. Più modelli di directory virtuale supportano lo sviluppo di applicazioni J2EE che hanno multiple directory virtuali. In tali applicazioni, il sito web risultante viene fisicamente diviso in directory senza radice comune. Ad esempio, in un negozio di vendita al dettaglio online www.mystore.com viene usato per il catalogo shopping, mentre order.mystore.com per

[22] Nel caso di un progetto che utilizza la procedura guidata per creare XDE, la sotto-directory del progetto viene creata automaticamente e la proprietà principale di origine del modello di codice viene impostata automaticamente.

controllare lo stato dell'ordine.

La proprietà principale di origine del modello di directory virtuale deve essere impostato alla directory in cui le risorse web da distribuire al contenitore web vengono poste. La proprietà principale di origine del modello di directory virtuale può essere impostato ad esempio alla sotto-directory “contenuto web” del progetto web[23].

Un esempio di modello di directory virtuale viene mostrato nella figura 22.

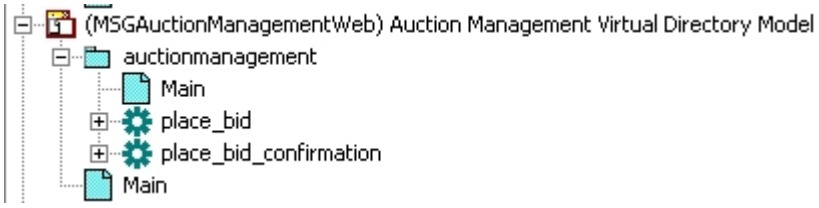


Figura 22: struttura modello di directory virtuale

Il modello di directory virtuale mostrato nella Figura 22 contiene risorse di codice sorgente diverse da Java che implementano gli elementi di progettazione della presentazione della gestione d'asta. È l'omologo dell'implementazione del codice sorgente diverso da Java al pacchetto “gestione d'asta” del pacchetto del livello “presentazione” nel “modello di progettazione globale” discusso nella sezione livelli di progettazione.

In questo esempio, la struttura del modello di directory virtuale riflette la struttura del “modello di progettazione globale” (discussa nella sezione livelli di progettazione). È presente un pacchetto per ogni pacchetto nel “modello di progettazione globale” il cui contenuto viene implementato da elementi diversi da Java da distribuire nel contenitore web, come JSP e pagine HTML. A causa dei limiti imposti dalla denominazione delle directory a cui ha accesso il server web, i nomi delle directory sotto quella virtuale non sono sempre uguali a quelli nel “modello di progettazione globale”.

Come mostrato nella Figura 22, il modello di directory virtuale contiene la rappresentazione visiva delle classi che specificano gli elementi d'implementazione. Queste classi rappresentano le classi di progettazione di RUP che si sono evolute fino al punto in cui è possibile implementarle e, nel caso di XDE, assegnarle un round-trip engineering. Diversamente, per il modello di codice EJB ed il modello di codice Java del progetto web, XDE non produce una rappresentazione visiva dei file di codice di origine associati nel modello di directory virtuale. Il nome del file di codice di origine associato viene mantenuto come proprietà della classe.

10. Deployment Model (Modello di distribuzione)

Il modello di distribuzione di RUP viene rappresentato da multipli modelli XDE - il “modello di distribuzione EAR” e una serie di modelli di distribuzione di XDE individuali, uno per ogni progetto XDE che contiene elementi da distribuire. Utilizzo dell'automazione della distribuzione relativa ai leverage XDE dei modelli di distribuzione multipla (XDE fornisce automazione per la creazione e la rifinitura di file di archivio J2EE e di descrittori di distribuzione oltre alla sincronizzazione di questi elementi di modello con quelli la cui origine è conservata nell'archivio).

Questa sezione fornisce esempi per i seguenti modelli di distribuzione:

- (Progetto di applicazione) modello di distribuzione EAR (vedere la sezione modello di sviluppo EAR)
- (Progetto EJB) modello di distribuzione EJB (vedere la sezione modello di distribuzione EJB)
- (Progetto web) modello di distribuzione web (vedere la sezione modello di distribuzione web)

Di seguito alcuni elementi generali senza valore rispetto alla distribuzione in XDE:

 I descrittori di distribuzione non vengono rappresentati in modo esplicito nei modelli XDE come file (artefatti UML). Vengono invece rappresentati dai contenuti dei modelli di distribuzione XDE. XDE utilizza gli elementi contenuti nei modelli di distribuzioneed i valori della proprietà assegnati a questi elementi, per determinare le informazioni sul contenuto scritto nei file del descrittore della distribuzione del modello.

[23] Nel caso di un progetto che utilizza la procedura guidata per creare XDE, la sotto-directory del progetto viene creata automaticamente e la proprietà principale di origne del modello di directory virtuale viene impostata automaticamente.

 XDE utilizza un modello di distribuzione EAR per tutte le distribuzioni, anche quelle di un singolo JAR o WAR di EJB. Questo riflette un limite della maggior parte dei server di applicazione, che richiedono un EAR per tutte le distribuzioni. Pertanto, anche se viene modellato solo JAR o WAR di EJB, è sempre necessario usare un modello di distribuzione EAR per i server di applicazione supportati da XDE. Tuttavia, XDE può esportare ogni tipo di archivio al file system. Questa abilità può essere utilizzata per la distribuzione ai server di applicazione che XDE non supporta. Dopo aver esportato l'archivio, è possibile invocare i tool specifici del server per completare la distribuzione.

 Diversi file di archivio J2EE possono essere definiti per differenti ambienti di distribuzione (verifica, produzione, ecc.). Questi archivi possono essere definiti nello stesso o in altri (separati) modelli di distribuzione XDE. L'automazione di XDE supporta entrambi, ma non definisce modelli di distribuzione predefiniti per progetti ed archivi predefiniti per ogni modello, tuttavia è possibile modificare queste definizioni, poiché vengono modellate nei modelli di distribuzione XDE. Qualunque sia l'approccio scelto, i modelli di distribuzione EJB devono essere definiti nei progetti EJB ed i modelli di distribuzione web nei progetti web().[24]

10.1 Modello di distribuzione EAR

Il “modello di distribuzione EAR” contiene la rappresentazione visiva del file di archivio dell'applicazione J2EE (file .EAR), informazioni del descrittore di distribuzione EAR e il nodo su cui EAR deve essere distribuito. Il “modello di distribuzione EAR” può anche contenere diagrammi che mostrano quali moduli J2EE (da altri modelli di distribuzione) sono contenuti all'interno di EAR.

Il “modello di distribuzione EAR” può anche essere utilizzato per descrivere la configurazione della distribuzione dell'applicazione nel suo complesso, così come la vista di distribuzione della struttura. Il “modello di distribuzione EAR” può contenere diagrammi che mostrano tutti i nodi di distribuzione e le relative connessioni, così come archivi specifici da distribuire a nodi specifici. Questi diagrammi devono far riferimento ad elementi (nodi e archivi) da tutti i modelli di distribuzione individuali.

Un esempio di un “modello di distribuzione” viene mostrato nella figura 23.

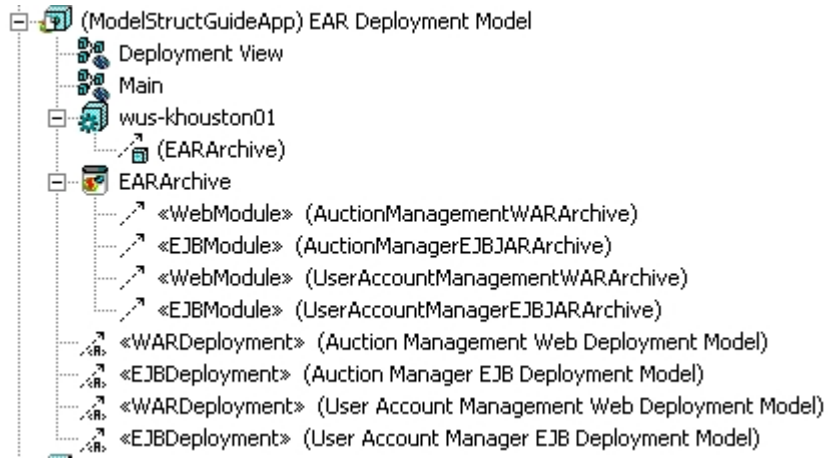


Figura 23: modello di distribuzione EAR

Il diagramma di “vista di distribuzione” rappresenta la vista di distribuzione dell'architettura. Include i nodi di distribuzione, le loro interconnessioni e gli archivi J2EE da distribuire a questi nodi. In alcuni casi, questo diagramma può includere esclusivamente l'archivio di applicazione J2EE e il nodo del server di applicazione su cui deve essere distribuito. Tuttavia, nel caso in cui archivi di modulo J2EE autonomi vengano distribuiti a nodi specifici, questo diagramma deve mostrare quali di questi specifici archivi vengono distribuiti a specifici nodi. Per informazioni sulle viste strutturali, vedere RUP.

[24] Un modello di distribuzione EJB deve essere in un progetto EJB, ma non deve essere lo stesso progetto come nel caso del corrispondente modello di codice EJB. Infatti, è possibile "mescolare e confrontare" gli EJB da differenti modelli di codice in un modello di distribuzione. Viene applicato qualche commento ai modelli web.

10.2 Modello di distribuzione EJB

Il modello di distribuzione EJB contiene componenti EJB, l'artefatto JAR di EJB e le informazioni del descrittore della distribuzione di EJB. Può anche contenere diagrammi che mostrano quali elementi di implementazione (dal modello di codice EJB) sono contenuti nel JAR di EJB. Per essere precisi, in questo modello, i componenti EJB vengono usati per rappresentare gli elementi d'implementazione e questi componenti vengono mappati al JAR di EJB.

Un esempio di modello di distribuzione EJB viene mostrato nella figura 24.

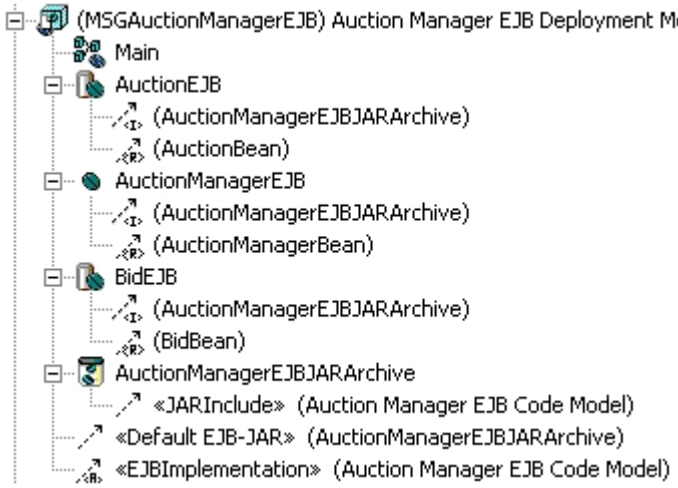


Figura 24: esempio di modello di distribuzione EJB

10.3 Modello di distribuzione web

Il modello di distribuzione web contiene componenti web, il file di archivio WAR e le informazioni del descrittore della distribuzione web. Il modello di distribuzione web può anche contenere diagrammi che mostrano quali elementi d'implementazione (dai modelli round-trip del progetto web) sono contenuti in WAR. Per essere precisi, in questo modello, i componenti Web vengono usati per rappresentare gli elementi d'implementazione e questi componenti vengono mappati al WAR.

Un esempio di modello di distribuzione web viene mostrato nella figura 25.

Figura 25: modello di distribuzione web

Rational
the software development company

Sedi:
Rational Software
18880 Homestead Road
Cupertino, CA 95014
Tel: (408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
Tel: (781) 676-2400

numero verde: (800) 728-1212
E-mail: info@rational.com
Web: www.rational.com
Ubicazione internazionale: www.rational.com/worldwide