

Linee guida sulla struttura del modello Rational
XDE™ per Microsoft« .NET

Indice

1. Introduzione	4
2. Scopo	4
3. Struttura del progetto XDE	4
4. Mappatura del modello RUP al modello XDE	7
5. Modello di caso d'uso	8
6. Modello di analisi	9
7. Modello di progettazione	11
7.1 Livelli di progettazione	12
7.2 Sottosistemi di progettazione	13
7.2.1 Specifica di sottosistema	13
7.2.2 Realizzazione di sottosistema	14
7.3 Realizzazioni di caso d'uso di progettazione	15
8. Modello di dati	15
8.1 Modello logico di dati (facoltativo)	15
8.2 Modello fisico di dati	16
8.3 Modello di dominio (facoltativo)	18
9. Modello di implementazione	19
9.1 Modello di codice C# per progetto di libreria di classi	20
9.2 Modello di codice C# per progetto di applicazione Web	20
10. Modello di distribuzione	21

1. Introduzione

In questo documento vengono fornite le raccomandazioni su come presentare e strutturare gli artefatti del modello RUP in Rational XDE™, Microsoft® .NET Edition. Naturalmente, che si decida o meno di modellare questi artefatti di RUP in XDE, si tratta di una decisione specifica per il progetto. All'interno di questo documento, si sottolinea per quali modelli viene fornito il supporto di automazione da parte di XDE e per quali non viene fornito, cosa che può influire sulla vostra decisione.

Poiché tutti i modelli XDE sono all'interno di progetti XDE, la sezione [Struttura del progetto XDE](#) fornisce le raccomandazioni su quali progetti XDE devono essere creati e quali file del modello XDE devono essere creati in tali progetti.

Sia RUP che XDE utilizzano il termine “modello” e la mappatura tra i modelli RUP e i modelli XDE non è sempre uno a uno. Nella sezione [Mappatura del modello RUP al modello XDE](#) viene descritta la mappatura dei relativi modelli.

La struttura di ciascuno degli artefatti del modello RUP nei relativi file del modello XDE viene descritta nella relativa sezione.

2. Scopo

Questo documento si concentra sulla descrizione delle strutture del file del modello XDE, non sul processo per lo sviluppo dei contenuti degli artefatti RUP associati. Questo documento non descrive anche l'euristica dettagliata per la definizione dei progetti XDE che contengono i modelli XDE descritti. Per informazioni su come definire, sviluppare e modellare i contenuti degli artefatti RUP, consultare RUP. Per ulteriori informazioni sui progetti, consultare la documentazione IDE.

Questo documento non descrive un esempio completo, ma utilizza invece esempi selezionati che mettono in risalto i punti da trattare; comunque, tutti gli esempi sono coerenti tra di loro e vengono presi dai modelli XDE attuali.

Le strutture di progetto e di modello descritte in questo documento sono solo raccomandazioni e potrebbero essere sostituite da qualsiasi numero di strutture ugualmente valide.

3. Struttura del progetto XDE

Questo documento si concentra 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 sulla struttura del progetto in cui si trovano le strutture di modello raccomandate.

Una soluzione di VS.NET è una raccolta di progetti e all'interno di ogni progetto possono esservi uno o più file di modello XDE¹. Di conseguenza, la struttura del progetto influisce sul numero di file di modello che vengono creati e anche sul loro contenuto.

Un'applicazione di .NET Enterprise può essere composta da più progetti, in base a come viene strutturata l'applicazione. Ad esempio, se l'applicazione implementa un servizio Web di XML, un'interfaccia Windows e Web, si raccomanda di disporre la soluzione di un servizio Web e di progetti di applicazione Windows e applicazione Web, rispettivamente. Per ulteriori informazioni sulle differenti maschere di progetto VS.NET, consultare la guida di VS.NET.

Per un'applicazione di .Net enterprise che viene sviluppata da più persone, si consiglia di creare i seguenti progetti e modelli XDE

¹

XDE definisce due tipi di file di modello: file di modello di codice e di non-codice. I file di modello di codice vengono utilizzati per modellare gli elementi specifici di linguaggio C# di un progetto mentre i file di modello di non-codice non dispongono di alcuna mappatura a un linguaggio di implementazione e funzionano da modelli di analisi e di progettazione. Esiste un solo file di modello di codice associato a un progetto mentre possono essere associati più file di modello di non codice.

Progetto XDE	Descrizione	Modelli XDE “?<nome di modello raccomandato>” (<Tipo di modello XDE: maschera di modello>]
Progetto di applicazione (progetto di modellazione di base di XDE)	Il progetto di applicazione rappresenta l'intera applicazione. Contiene i file di modello XDE che descrivono l'applicazione come un insieme	<ul style="list-style-type: none"> - “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 di implementazione globale” (Rational XDE: Modello vuoto) - “Modello di distribuzione” (.Net: Modello di distribuzione)
Progetto di modellazione dei dati (Progetto di modellazione dei dati di XDE)	Il progetto di modellazione dei dati contiene le risorse necessarie per modellare i dati dell'applicazione e per la progettazione roundtrip di un modello di dati a/da un database.	<ul style="list-style-type: none"> - “Modello logico dei dati” (Dati: modello logico dei dati) - “Modello fisico dei dati2”(Dati: file di modello fisico dei dati specifici del fornitore) 2 - “Modello di dominio” (Dati: file di modello di dominio specifico del fornitore)
Progetti della libreria di classi di .Net (progetto di modellazione di libreria di classi)	<p>Un progetto di libreria di classe è un progetto “C#” di VS.NET che contiene elementi C# (cioè, classi, interfacce ecc.) necessarie per implementare una libreria di classi (Gestione aste in questo esempio). Gli elementi contenuti vengono impacchettati e distribuiti come un gruppo di .NET.</p> <p>Si noti che un'applicazione di .NET può contenere più di un progetto di libreria di classi come questo. Come indicato in precedenza, la selezione del numero di progetti è una scelta strutturale e può variare per differenti applicazioni.</p>	<ul style="list-style-type: none"> - “Modello di codice di .Net” (.Net: modello di codice di .Net) - “Modello di distribuzione di .Net” (.Net: modello di distribuzione di .Net)
Progetti Web (Progetto di modellazione Web di XDE)	<p>I progetti Web rappresentano le risorse Web dell'applicazione. Gli elementi contenuti vengono impacchettati e distribuiti in un gruppo di .Net.</p> <p>I progetti Web distinti possono essere definiti per aree specifiche della logica di presentazione. Si consiglia di creare un progetto Web per ogni gruppo che deve essere prodotto. Se si definiscono progetti distinti, il nome del progetto deve riflettere i suoi contenuti.</p>	<ul style="list-style-type: none"> - “Modello di codice di .Net” (.Net: modello di codice di .Net) - “Modello di distribuzione di .Net” (.Net: modello di distribuzione di .Net)

Nella figura 1 viene mostrato un esempio di questa organizzazione di progetto e di modello che utilizza le viste di Solutions Explorer e XDE Model Explorer. La vista di Solution Explorer viene mostrata sul lato sinistro della figura mentre quella di Model Explorer sul lato destro.

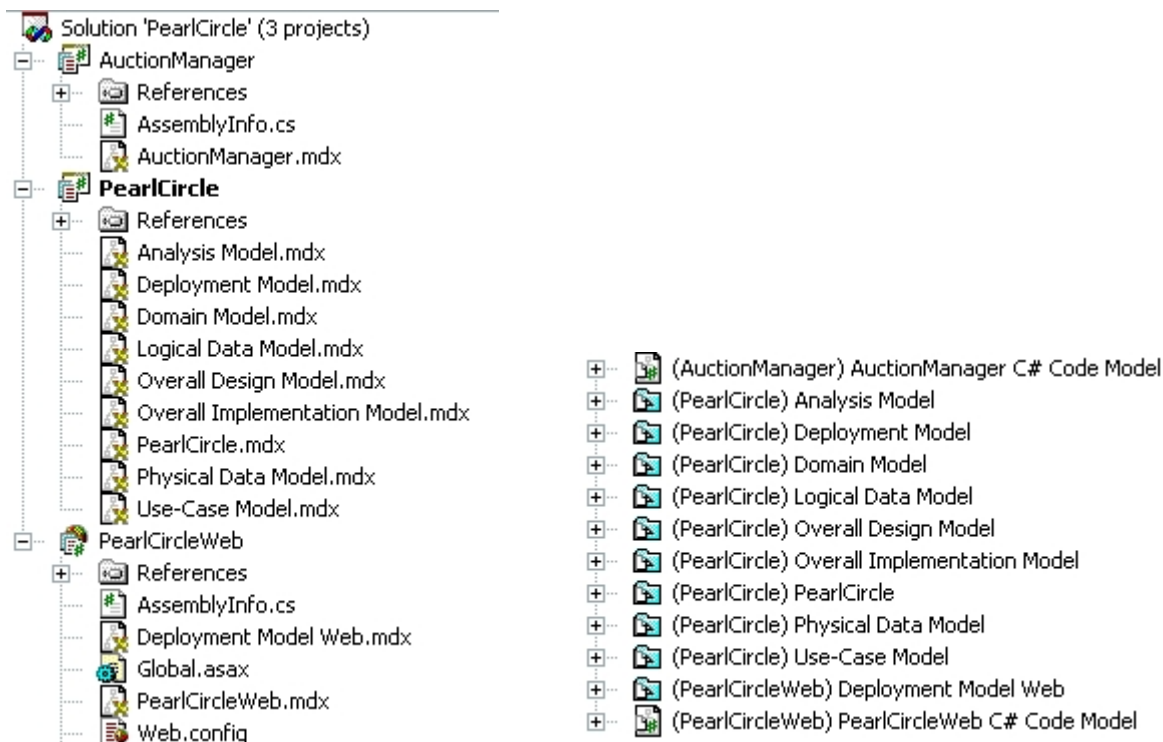


Figura 1: Modello e organizzazione di progetto

In alternativa, se l'applicazione è davvero piccola e viene sviluppata da una singola persona, la struttura di progetto di cui sopra può essere semplificata in due progetti: uno che contiene l'intera applicazione e gli elementi non Web e l'altro che contiene gli elementi Web. Oltre alla riduzione del numero di progetti, è possibile ridurre anche il numero di modelli allo stesso modo. Ad esempio, per un piccolo progetto da singola persona, sono possibili le seguenti semplificazioni:

- ☐ Non viene mantenuto un modello di analisi separato. Sia l'analisi che la progettazione vengono eseguiti nei modelli roundtrip XDE.
- ☐ Non viene mantenuto un “modello di progettazione globale” né un “modello di implementazione globale”. Il progetto è sufficientemente piccolo per cui sia possibile ottenere una panoramica osservando direttamente i modelli roundtrip XDE. Inoltre, le realizzazioni di casi d'uso vengono mantenute nel modello di codice .Net.
- ☐ Non viene mantenuto un modello logico di dati separato. Uno schema fisico di dati viene sviluppato direttamente nel “modello fisico di dati”.

Questa “piccola struttura di progetto” viene riassunta nella successiva tabella.

Progetto XDE	Descrizione	Modelli XDE “<nome di modello raccomandato>” (<Tipo di file XDE: maschera di modello>]
Progetto di applicazione (progetto di modellazione di libreria XDE)	Il progetto di applicazione rappresenta gli aspetti non-Web dell'applicazione. Contiene i modelli di classi che descrivono l'applicazione come un insieme, il modello di dati e i modelli specifici di .Net.	<ul style="list-style-type: none"> - “Modello di caso d'uso” (Rational XDE: modello di caso d'uso) - “Modello fisico di dati” (Dati: file di modello fisico di dati specifico per il fornitore) - “Modello di codice .Net” (.Net: modello di codice .Net) - “modello di distribuzione .Net” (.Net: modello di distribuzione .Net)
Progetto Web (Progetto di	I progetti Web rappresentano le risorse Web dell'applicazione. Gli	<ul style="list-style-type: none"> - “Modello di codice .Net” (.Net: modello di codice .Net) “Modello di distribuzione .Net” (.Net: modello di distribuzione

modellazione Web XDE)	elementi contenuti vengono impacchettati e distribuiti in un gruppo .Net.	.Net)
-----------------------	---	-------

Nella figura 2 viene mostrato un esempio di una piccola organizzazione di progetto e di modello.

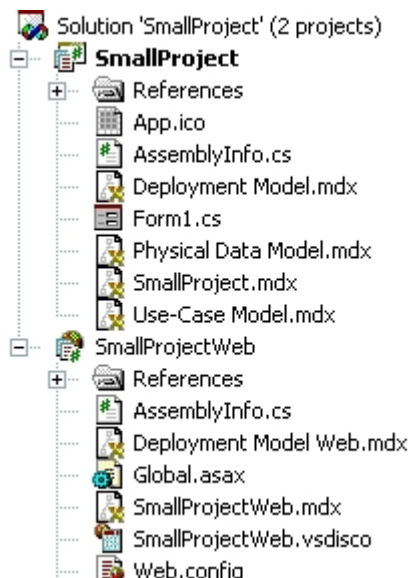


Figura 2: Esempio di piccola organizzazione di progetto e di modello XDE

La selezione attuale del numero di progetti e dei diversi file di modello è una scelta strutturale e potrebbe variare per progetti diversi. Comunque, non ha importanza quanti siano i progetti definiti, poiché può esserci solo un file di modello di codice .Net di XDE per progetto. Per ulteriori informazioni sui progetti e sui file di modello XDE che possono contenere, consultare la documentazione di XDE.

Si raccomanda in modo particolare che i nomi di modello XDE siano univoci attraverso tutti i progetti XDE. Ciò è estremamente importante quando si tenta di risolvere i riferimenti tra modelli XDE. Per ulteriori informazioni sui riferimenti tra modelli e sulla loro risoluzione, consultare la documentazione di XDE.

La struttura dei modelli XDE mostrata nella figura 1 costituisce il punto di convergenza per il resto del documento.

4. Mappatura del modello RUP al modello XDE

Prima di descrivere come rappresentare gli artefatti del modello RUP in XDE, è importante occuparsi della confusione tra "modello RUP" e "modello XDE" poiché sono cose diverse e la mappatura dai modelli RUP ai modelli XDE associati non è sempre uno a uno (chiusa, ma non uno a uno). Poiché viene utilizzata la parola "modello" sia per RUP che per XDE, la presunzione iniziale è che siano gli stessi. Tuttavia, i modelli in RUP separano le attività di processo (analisi, progettazione, implementazione ecc.), mentre i modelli in XDE separano le attività di sviluppo (modelli di codice separati per la descrizione della struttura di impacchettamento del linguaggio di programmazione rispetto a una struttura virtuale di directory, modelli di codice separati per differenti linguaggi di programmazione e gli ambienti di sviluppo ecc.).

Per diminuire tale confusione, nel contesto di questo white paper il termine "modello" viene esplicitamente qualificato con "RUP" o "XDE".

La successiva tabella riassume la mappatura da modello RUP a modello XDE. I modelli XDE sono quelli introdotti nella sezione [Struttura di progetto XDE](#). La struttura di ciascuno dei modelli XDE viene descritta nelle sezioni successive di questo white paper.

Modello RUP	<Progetto XDE>: < Nome modello XDE>
Modello di 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
Modello di dati	Modelli di dati XDE Progetto di modellazione dati: modello logico di dati Progetto di modellazione dati: modello fisico di dati specifico per il fornitore Progetto di modellazione dati: modello di dominio specifico per il fornitore
Modello di implementazione	Progetto di applicazione: modello di implementazione
Modello di distribuzione	Progetto di applicazione: modello di distribuzione

5. Modello di caso d'uso

La struttura consigliata del “modello di caso d'uso” è mostrata nella figura 3.

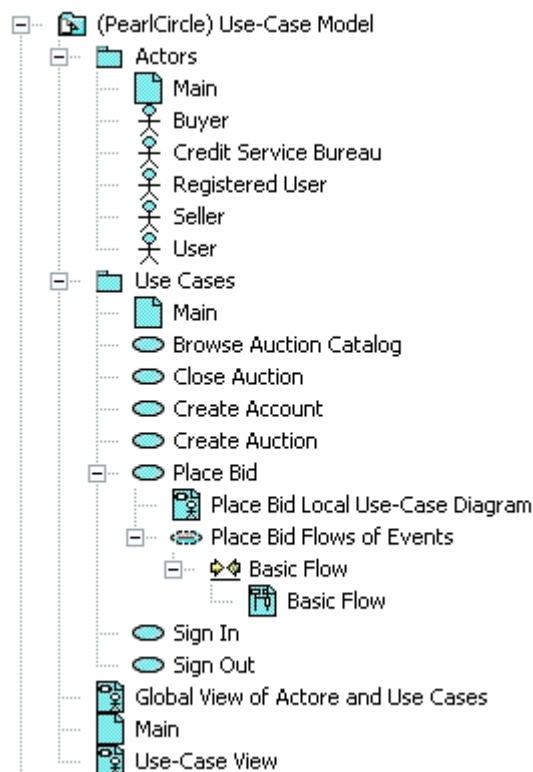


Figura 3: Struttura di "modello di caso d'uso"

Il “modello di caso d'uso” è diviso in due pacchetti: “attori” e “casi d'uso”.

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

- ☐ Il diagramma “Diagramma di caso d'uso Posiziona offerta” contiene il caso d'uso “Posiziona offerta” e gli attori che partecipano in quel caso d'uso.
- ☐ L'istanza di collaborazione “Flusso di eventi di Posiziona offerta” contiene i diagrammi di interazione che descrivono

graficamente i flussi di eventi descritti nella descrizione del caso d'uso (cioè, le interazioni tra gli attori e il caso d'uso). Tale istanza di collaborazione non deve essere confusa con le Realizzazioni di caso d'uso, descritte sia nella sezione Modello di analisi che nella sezione Realizzazioni di casi d'uso di progettazione, poiché le istanze di collaborazione nel “modello di caso d'uso” sono rigorosamente “black box” e non descrivono le interazioni di elementi all'interno dell'applicazione.

- Il grafico dell'attività “Flusso di eventi di Posiziona offerta” contiene diagrammi di attività che descrivono graficamente il flusso degli eventi descritti nella descrizione di caso d'uso.

Nell'esempio mostrato in figura 3, il diagramma “Vista globale di attori e casi d'uso” contiene tutti i casi d'uso e gli attori insieme alle loro relazioni, a differenza dei diagrammi “Principale”, che contengono gli elementi nei pacchetti in cui sono presenti i diagrammi “Principale”. Se sono presenti molti attori e casi d'uso, le informazioni sul diagramma “Vista globale di attori e casi d'uso” possono essere espresse utilizzando più diagrammi.

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

Se desiderato, è possibile creare pacchetti aggiuntivi nei pacchetti di attori e casi d'uso per organizzare ulteriormente gli elementi di modello contenuti, come mostrato in figura 4.

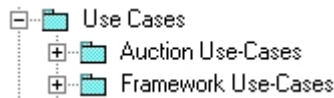


Figura 4: Suddivisione del pacchetto aggiuntivo di casi d'uso

6. Modello di analisi

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

Nota: la decisione se mantenere o meno un modello di analisi e un modello di progettazione separati è specifica del progetto. Se un modello di analisi separata non viene mantenuto, allora le classi di analisi verranno spostate nell'appropriata partizione del modello di progettazione³ e perfezionate. Un'altra opzione è di creare semplicemente le classi di analisi e le realizzazioni di caso d'uso di analisi nel modello di progettazione e quindi svilupparle nella loro forma di progettazione. Consultare 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 è mostrata nella figura 5.

³

Come si vedrà più avanti, la “partizione di modello di progettazione globale” adatta potrebbe essere proprio un pacchetto di uno dei modelli roundtrip XDE, poiché la progettazione degli elementi specifici della tecnologia viene eseguita nei modelli roundtrip.

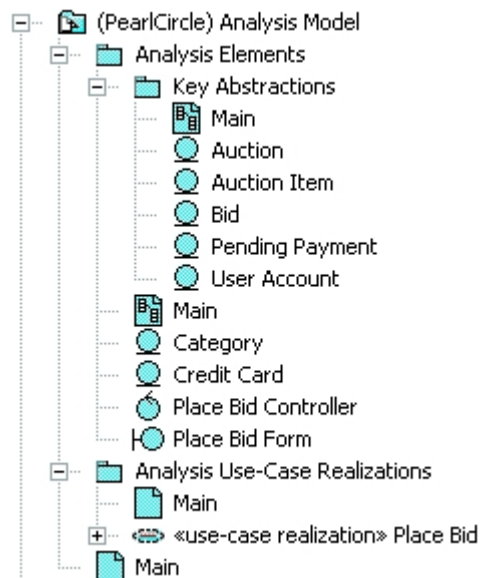


Figura 5: Struttura di modello di analisi

Il pacchetto “Elementi di analisi” contiene le Classi di analisi. Le istanze delle Classi di analisi vengono visualizzate nei diagrammi del pacchetto “Realizzazioni di caso d'uso di analisi”.

In aggiunta alle Classi di analisi, è possibile definire i pacchetti nel pacchetto “Elementi di analisi” per un'ulteriore suddivisione delle Classi di analisi contenute (vedere il pacchetto “Astrazioni chiavi” in figura 5). Tale suddivisione aggiuntiva è facoltativa, soprattutto se non viene mantenuto un Modello di analisi separato. In tali casi, le Classi di analisi possono essere considerate “transitorie” (cioè, esistono solo fino a quando non si evolvono in elementi di progettazione), per cui la loro organizzazione non viene considerata critica. Una possibile eccezione sono le Classi di analisi di astrazione chiave.

Come mostrato in figura 5, il pacchetto di “Astrazioni chiave” contiene le Classi di analisi che vengono considerate per rappresentare le astrazioni chiave del sistema. Come indicato in precedenza, questo pacchetto è facoltativo. Un'alternativa è quella di rappresentare le astrazioni chiave su un diagramma di classi nel pacchetto “Elementi di analisi”. Comunque, la creazione di un pacchetto separato fornisce una classificazione più esplicita delle Classi di analisi come astrazioni chiave. Di fatto, anche se un modello di analisi non viene mantenuto nella sua interezza, alcuni progetti possono scegliere di mantenere le classi di analisi di astrazione chiave. In tali casi, è utile la definizione di un pacchetto separato per contenere le Classi di analisi che vengono mantenute.

Nota: le astrazioni chiave vengono mostrate anche sul diagramma “Vista logica: astrazioni chiave” nel “Modello di progettazione globale”. Per ulteriori informazioni, consultare la sezione Modello di progettazione.

Il pacchetto “Realizzazioni di caso d'uso di analisi” contiene le Realizzazioni di caso d'uso a livello di analisi, e descrivono in che modo vengono eseguiti i Casi d'uso in termini di Classi di analisi del pacchetto “Elementi di analisi”. Ognuna delle Realizzazioni di caso d'uso di analisi che realizza un Caso d'uso nel Modello di caso d'uso, ha lo stesso nome del Caso d'uso e dovrebbe avere la struttura come quella mostrata in figura 6.

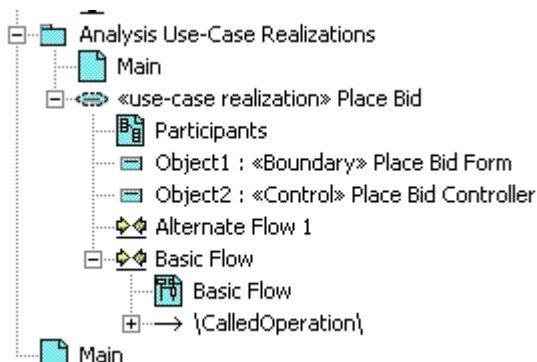


Figura 6: Struttura del pacchetto “Realizzazione di caso d'uso di analisi”

Il diagramma “Partecipanti” mostra le Classi di analisi (dal pacchetto “Elementi di analisi”) che partecipano nella Realizzazione di caso d'uso (cioè, quelle Classi di analisi le cui istanze vengono mostrate sui diagrammi di interazione) e le relazioni che supportano la collaborazione descritta nei diagrammi di interazione.

Le istanze di interazione di "flusso" ("Flusso di base" e "Flusso alternato 1") contengono diagrammi di sequenza che descrivono i flussi di eventi di Caso d'uso. Dovrebbe esistere una istanza di interazione per ogni flusso di eventi di caso d'uso significativo. I diagrammi di sequenza nelle istanze di interazione descrivono il flusso tra le Classi di analisi partecipanti durante l'esecuzione del caso d'uso associato.

7. Modello di progettazione

Il Modello di progettazione RUP viene rappresentato da più modelli XDE: il “Modello di progettazione globale” e gli elementi di progettazione sottoposti a roundtrip che risiedono in modelli roundtrip XDE separati (gli elementi di progettazione sottoposti a roundtrip sono elementi di progettazione dettagliati che partecipano all'ingegnerizzazione roundtrip). Così, l'automazione disponibile nei singoli modelli roundtrip può essere influenzata.

Il “Modello di progettazione globale” descrive la progettazione dell'applicazione come un insieme e contiene elementi che attraversano più modelli roundtrip. Contiene partizioni logiche che dettano l'organizzazione dei singoli modelli roundtrip, e Realizzazioni di caso d'uso che uniscono tutto insieme (le Realizzazioni di caso d'uso descrivono la collaborazione tra gli elementi di progettazione provenienti da diversi modelli roundtrip). Il “Modello di progettazione globale” contiene diagrammi che fanno riferimento a elementi di progettazione sottoposti a roundtrip. Per informazioni sui singoli modelli roundtrip XDE, vedere la sezione Modello di implementazione.

Un'altra possibilità è quella di rappresentare il Modello di progettazione e il Modello di implementazione nello stesso modello di codice XDE. Ciò è possibile solo se si dispone di un linguaggio di implementazione mirato e il proprio team è piccolo.

Il mantenimento del “Modello di progettazione globale” è facoltativo, ma può essere una buona idea per organizzare i diagrammi, aumentare il livello di astrazione ecc., e anche per fornire un posto per gli elementi di progettazione mentre si determina quale meccanismo di implementazione applicare.

La struttura raccomandata del “Modello di progettazione globale” è mostrata in figura 7.

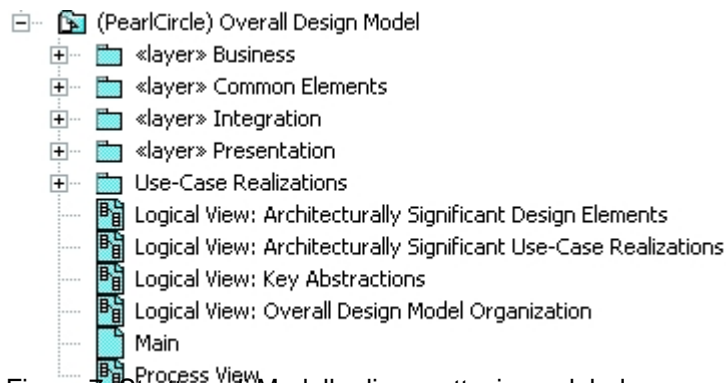


Figura 7: Struttura di Modello di progettazione globale

Questo Modello di progettazione globale contiene i seguenti pacchetti:

- I pacchetti di 1/2livello η contengono (oppure contengono diagrammi che li riportano) gli elementi di progettazione del sistema (classi di progettazione, interfacce e sottosistemi di progettazione). Questa struttura rappresenta una particolare strategia di suddivisione che viene descritta nella sezione Livelli di progettazione.
- Il pacchetto “Realizzazioni di caso d'uso” contiene Realizzazioni di caso d'uso a livello di progettazione. La struttura interna delle Realizzazioni di caso d'uso viene discussa in modo più dettagliato nella sezione Realizzazione di caso d'uso di progettazione.

I diagrammi che rappresentano viste strutturali includono il termine “Vista” nel nome del diagramma. Per ulteriori informazioni sulle viste strutturali, consultare RUP.

Il diagramma “Vista logica: astrazioni chiave” contiene le astrazioni chiave del sistema. Esistono diverse opzioni per mantenere queste astrazioni chiave:

- Viene mantenuto un **Modello di analisi completo**. In questo caso, il diagramma “Vista logica: astrazioni chiave” contiene le **Classi di analisi** provenienti dal **Modello di analisi** che rappresentano le astrazioni chiave del sistema.
- Viene mantenuto un **Modello di analisi parziale**, vale a dire, solo le astrazioni chiave. In questo caso, il diagramma “Vista logica: astrazioni chiave” contiene le **Classi di analisi** provenienti dal **Modello di analisi** che rappresentano le astrazioni chiave del sistema.
- Non viene mantenuta alcuna parte del **Modello di analisi**. In questo caso, le **Classi di analisi** che rappresentano le astrazioni chiave possono essere mantenute in un pacchetto del 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 1° livello contengono gli elementi di progettazione del sistema (cioè, **Classi di progettazione**, **Interfacce** e **Sottosistemi di progettazione**) che si sviluppano dalle **Classi di analisi**. I pacchetti di 1° livello possono contenere qualsiasi numero di pacchetti secondari che in seguito suddividono gli elementi di progettazione contenuti. Le Realizzazioni di caso d'uso di progettazione (contenute nel pacchetto “Realizzazioni di caso d'uso” del “Modello di progettazione globale” vengono discusse dopo l'intestazione della sezione [Realizzazioni di caso d'uso di progettazione](#)) vengono scritte in termini di elementi di progettazione contenuti in tali pacchetti. Il **Modello di progettazione** può seguire qualsiasi numero di strategie di suddivisione. La strategia di suddivisione descritta in questa sezione, viene mostrata nella figura 8.

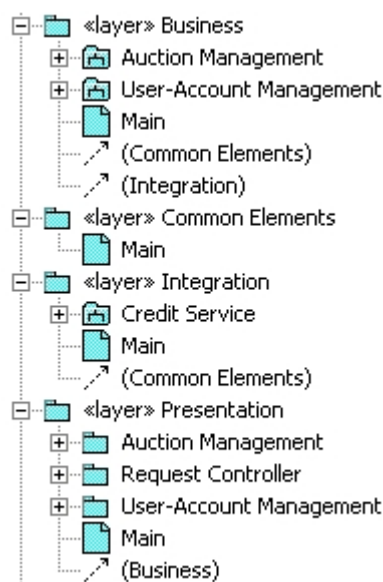


Figura 8: Esempio di suddivisione del pacchetto di progettazione

In questo esempio, i pacchetti di primo livello vengono considerati livelli, dove ogni livello ha una specifica responsabilità. I pacchetti di secondo livello suddividono più avanti gli elementi del pacchetto di livello per funzionalità business. Il pacchetto di livello “Presentazione” è responsabile della gestione delle interazioni con l'utente finale. In un'applicazione .Net, gli elementi di progettazione che potrebbero risiedere nel pacchetto di livello “Presentazione” comprendono Active Server Pages (ASP.NET). È possibile più avanti dividere il pacchetto di livello “Presentazione” in pacchetti secondari per raggruppare gli elementi che appartengono a un insieme di casi d'uso correlati; ad esempio, il pacchetto “Gestione aste” in figura 8. Il pacchetto di livello “Business” è responsabile dell'esecuzione di qualsiasi elaborazione business. Nella struttura “Modello di progettazione globale” presentata in questo documento, il pacchetto di livello “Business” comprende un insieme di pacchetti di progettazione secondari, uno per funzione business principale (ad esempio, la “Gestione aste” e la “Gestione account

utente”, pacchetti di sottosistema in figura 8). I pacchetti Sottosistema di progettazione vengono descritti in modo più dettagliato dopo l'intestazione della sezione [Sottosistemi di progettazione](#).

Il pacchetto di livello “Integrazione” è responsabile nel fornire accesso alle risorse di back-end, che includono i database e i sistemi esterni. Nella struttura Modello di progettazione presentata in questo documento, il pacchetto di livello “Integrazione” comprende anche i pacchetti di sottosistema di progettazione, uno per sistema esterno (ad esempio, il pacchetto di sottosistema “Servizio di accreditamento” in figura 8). I pacchetti Sottosistemi di progettazione vengono descritti in modo più dettagliato sotto l'intestazione della sezione [Sottosistemi di progettazione](#).

Il pacchetto di livello “Elementi comuni” contiene gli elementi che vengono condivisi tra i livelli.

Di nuovo, le strutture descritte in questa sezione possono essere sostituite con una struttura diversa che riflette una diversa strategia di suddivisione.

7.2 Sottosistemi di progettazione

I sottosistemi di progettazione vengono rappresentati dai pacchetti di sottosistema nel “Modello di progettazione globale”. Tutti i pacchetti di sottosistema di progettazione dovrebbero avere la stessa struttura. Le specifiche di tale struttura variano in funzione del livello di dettaglio acquisito per il Sottosistema di progettazione.

Un esempio di una struttura più formale e rigorosa di Sottosistema di progettazione è mostrata nella figura 9.

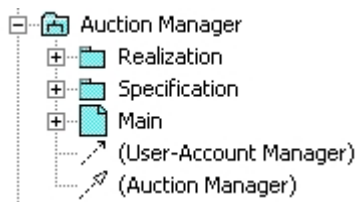


Figura 9: Struttura di sottosistema di progettazione

Tale struttura di pacchetto di sottosistema di progettazione supporta la definizione dei pacchetti “Specifica” e “Realizzazione” all'interno del pacchetto di sottosistema di progettazione. Tale struttura è stata influenzata dal libro *UML Components: A Simple Process for Specifying Component-Based Software* scritto da J. Cheesman and J. Daniels. È possibile utilizzare una struttura di pacchetto di sottosistema di progettazione semplificata che non contiene tali partizioni, senza avere impatto sulle altre strutture di file di modello definite in questo documento. Ciascuno dei pacchetti “Specifica” e “Realizzazione” viene discusso nelle successive sezioni.

7.2.1 Specifica di sottosistema

Il pacchetto “Specifica” contiene una descrizione delle interfacce di Sottosistema di progettazione.⁴ Un esempio di una specifica di sottosistema è mostrata nella figura 10.



Figura 10: Esempio di specifica di sottosistema di progettazione

⁴

In questo semplice esempio, può sorgere il dubbio sulla necessità di un pacchetto separato solo per l'interfaccia. Comunque in un progetto reale il pacchetto viene gestito meglio poiché può contenere riferimenti a documenti che descrivono il sottosistema e, in particolare, i vincoli di interfaccia come le condizioni preliminari e successive sulle operazioni.

7.2.2 Realizzazione di sottosistema

Il pacchetto “Realizzazione” contiene una descrizione di come viene realizzata la specifica di Sottosistema di progettazione . Un esempio di pacchetto “Realizzazione” di un pacchetto di sottosistema di progettazione è mostrato in figura 11.

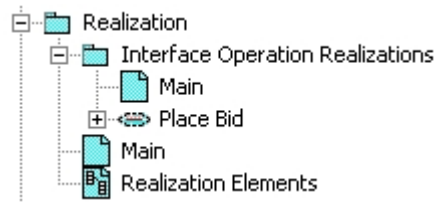


Figura 11: Esempio di realizzazione di sottosistema di progettazione

Il diagramma “Elementi di realizzazione” contiene riferimenti agli elementi di progettazione che realizzano il sottosistema. Gli stessi elementi di progettazione risiedono in un modello di codice .Net, dove partecipano all'ingegnerizzazione roundtrip. Per ulteriori informazioni, vedere la sezione Modello di implementazione.

Il pacchetto “Realizzazioni di operazione di interfaccia” contiene istanze di collaborazione che descrivono come gli elementidi sottosistema realizzano operazioni significative delle interfacce di Sottosistema di progettazione (nel pacchetto “Specifica”). Esiste una istanza di collaborazione per operazione di interfaccia di sottosistema significativa.s Un esempio di un pacchetto di “Realizzazioni di operazione di interfaccia” è mostrato nella figura 12.

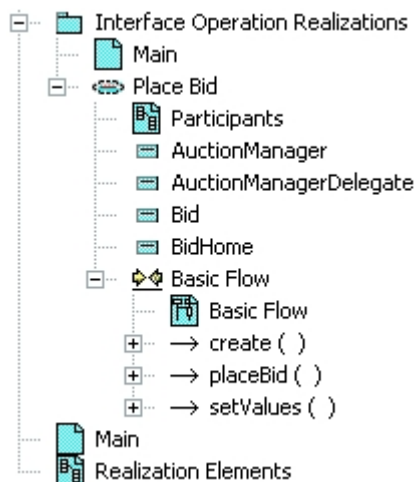


Figura 12: esempio di pacchetto di realizzazioni di operazione di interfaccia

Come con le Realizzazioni di caso d'uso a livello di analisi (discusse in precedenza nella sezione Modello di analisi) e con le Realizzazioni di caso d'uso a livello di progettazione (discusse più avanti nella sezione Realizzazioni di caso d'uso di progettazione), ogni realizzazione di operazione di interfaccia contiene un diagramma di classi contenente gli elementi di sottosistema che partecipano alla realizzazione (il diagramma “Partecipanti” in figura 12), come pure i diagrammi di interazione che descrivono in che modo tali partecipanti collaborano per eseguire l'operazione di interfaccia di sottosistema (il diagramma “Flusso di base” nella figura 12).

7.3 Realizzazioni di caso d'uso di progettazione

Il pacchetto “Realizzazioni di caso d'uso” contiene Realizzazioni di caso d'uso a livello di progettazione. Ognuna delle Realizzazioni di caso d'uso che viene associata a un caso d'uso nel Modello caso d'uso, ha lo stesso nome del Caso d'uso e dovrebbe avere la struttura mostrata in figura 13.

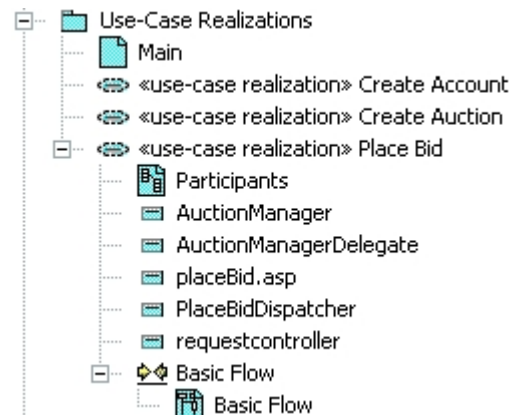


Figura 13: Struttura di realizzazione di caso d'uso di progettazione

Il diagramma “Partecipanti” di Realizzazione di caso d'uso ha una casella di testo alla classe code-behind XDE, verrà eseguito il forward engineering della variabile membro privata nella classe code-behind.

Il diagramma “Flusso di base” è un esempio di un programma di interazione che descrive il flusso tra gli elementi di progettazione partecipanti durante l'esecuzione del caso d'uso associato. Dovrebbe esserci una istanza di interazione per ogni flusso di eventi nel caso d'uso.

È importante notare che i diagrammi di Realizzazione di caso d'uso possono contenere (e di solito è così) riferimenti agli elementi di progettazione che risiedono fisicamente in modelli roundtrip XDE separati. Nella Realizzazione di caso d'uso viene dimostrata la collaborazione tra elementi in modelli roundtrip separati.

8. Modello di dati

Il Modello di dati di RUP viene rappresentato da più file di modello XDE:

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

La separazione dei file di modello XDE fornisce la flessibilità ottimale per l'automazione supportata tra il “Modello di progettazione globale”, il Modello di dati e il database fisico.

Di seguito vengono descritti tali file di modello XDE in modo più dettagliato.

8.1 Modello logico di dati (facoltativo)

Il modello logico di dati può essere utilizzato nelle situazioni dove il progetto ha bisogno di creare una rappresentazione logica dei dati autonoma delle entità fondamentali e delle relazioni importanti per la progettazione del database. La creazione di un modello logico di dati XDE è facoltativa perché il team di progettazione del database potrebbe invece trasformare le Classi di progettazione persistenti del Modello di progettazione in tabelle nel Modello di dati per creare la struttura di progettazione fisica iniziale del database direttamente nel

modello fisico di dati XDE (vedere la sezione in basso Modello fisico di dati).

Il modello logico di dati XDE può essere suddiviso in pacchetti di area di argomento, se necessario. I pacchetti di area di argomento definiscono raggruppamenti logici di classi di entità. Il modello logico di dati XDE può anche contenere un pacchetto “Elementi comuni” che contiene elementi di modello trasversali alle aree di argomento.

I diagrammi che hanno il termine “Vista” nel nome, vengono utilizzati per documentare la Vista dati dell'architettura. Il diagramma “Vista dati: Organizzazione di modello logico di dati” viene utilizzato per documentare l'organizzazione ad alto livello dei dati del modello logico di dati, come espresso nelle principali partizioni (cioè, pacchetti) del Modello logico di dati XDE. La “Vista dati: Elementi di dati logici chiave” viene utilizzata per documentare gli elementi logici chiave del Modello di dati. Se venisse mantenuto un Modello logico di dati (cioè, se esistesse un “Modello logico di dati” separato), allora tale diagramma conterrebbe elementi del Modello logico di dati XDE. Per ulteriori informazioni sulle viste di architettura, consultare RUP.

Nella figura 14 viene mostrato un esempio di struttura di Modello logico di dati XDE raccomandata.

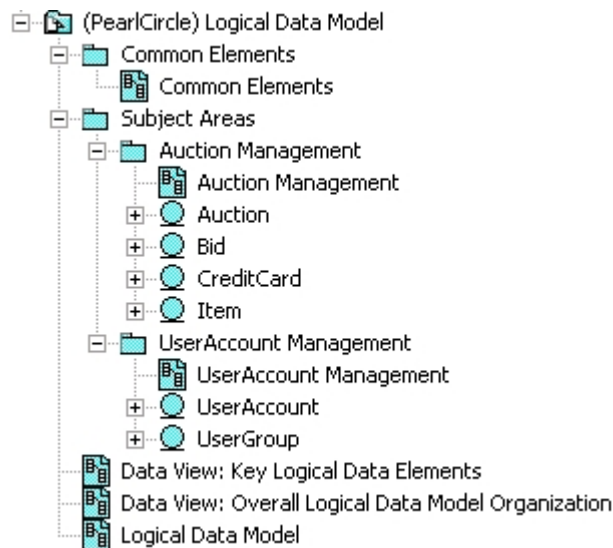


Figura 14: Struttura di modello logico di dati XDE

In questo esempio, ci sono due pacchetti di area di argomento, “Gestione aste” e “Gestione account utente”. Ogni pacchetto di area di argomento contiene le classi di entità che insieme comprendono il Modello logico di dati. Non esiste una mappatura diretta alle strutture di pacchetto nel Modello di progettazione sebbene possano esserci delle similitudini.

8.2 Modello fisico di dati

Il Modello fisico di dati contiene la tabella dettagliata di database e memorizza le progettazioni di procedura memorizzate che vengono utilizzate per implementare il database attraverso le risorse di forward engineering di XDE Data Modeler. Il Modello fisico di dati comprende anche elementi di modello utilizzati per definire la configurazione di memorizzazione fisica del database. In generale, gli elementi di modello comprendono i database e i tablespaces che comprendono il layout fisico delle tabelle del database sui supporti di archiviazione di destinazione.

Quando si crea un Modello fisico di dati XDE, il progettista del database deve selezionare il database di destinazione appropriato. I database supportati sono: DB2 MVS, DB2 UDB, Oracle, Sybase e SQL Server. XDE fornirà il nome del file del modello XDE al database selezionato. Nell'esempio di “Modello fisico di dati” di questo documento, il nome del file del modello XDE è stato aggiornato a “Modello fisico di dati”. Un progettista di database può scegliere di accettare il nome predefinito durante la creazione del “Modello fisico di dati”.

Nella figura 15 viene mostrato un esempio di struttura di Modello fisico di dati XDE raccomandata.

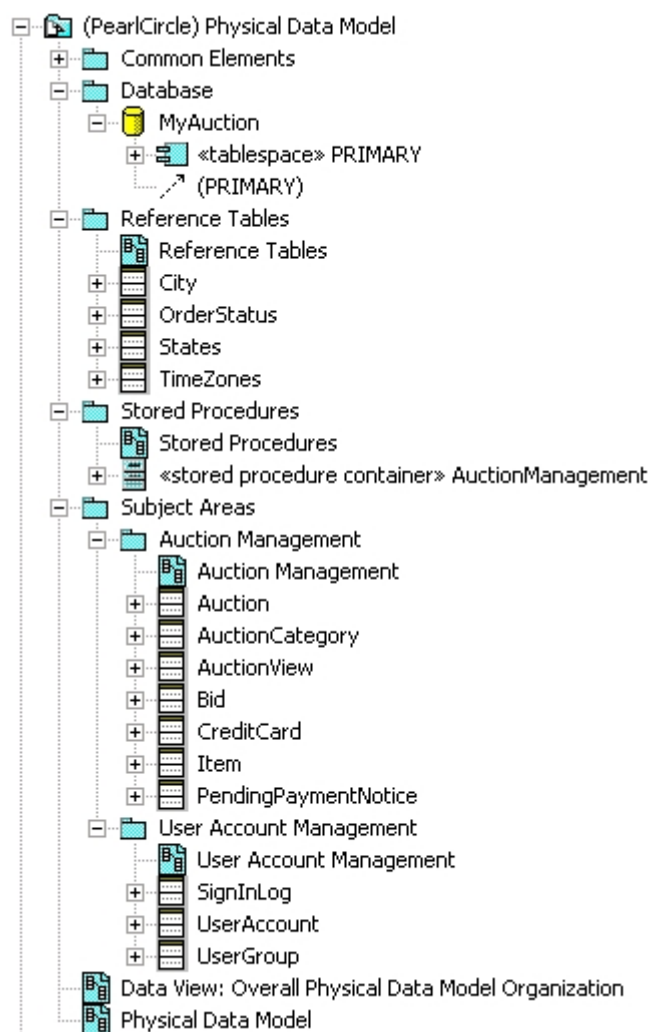


Figura 15: Struttura di Modello fisico di dati XDE

Il pacchetto “Elementi comuni” contiene le tabelle del database e le viste trasversali alle aree di argomento.

Il pacchetto “Database” contiene gli elementi di modello che definiscono la configurazione di memorizzazione fisica del database. Contiene i database e i tablespace che comprendono il layout fisico delle tabelle del database sui supporti di archiviazione di destinazione. I tablespace vengono utilizzati per raggruppare logicamente le tabelle all'interno di un database. Per le linee guida sulla definizione dei tablespace, consultare RUP. Il pacchetto “Database” può essere suddiviso in pacchetti di livello inferiore se necessario, in base alla complessità dell'applicazione.

Nell'esempio mostrato in figura 15, il pacchetto “Database” contiene un unico database, MyAuction, i relativi tablespace associati, PRIMARY, e le relazioni di realizzazione di tabella. Il tablespace può essere denominato con qualsiasi nome appropriato per un progetto di database. Per il database MyAuction, viene definito solo un tablespace, denominato PRIMARY. Quando si esegue il forward engineering, vengono create le tabelle collegate al database tramite relazione di realizzazione con il tablespace di database (sia in un database che in un DDL).

Il pacchetto “Tabelle di riferimento” contiene tabelle di dati statici che conservano informazioni di dati “costanti” necessarie per l'applicazione.

Il pacchetto “Procedure memorizzate” contiene tutte le classi che rappresentano le procedure memorizzate in database (le classi $\frac{1}{2}$ contenitore di procedura memorizzata e le operazioni associate di $\frac{1}{2}$ procedura memorizzata). Le procedure memorizzate che fanno riferimento a una singola tabella possono essere impacchettate sia nel pacchetto “Procedure memorizzate” che nel pacchetto “Area di argomento” con la

tabella a cui fa riferimento la procedura memorizzata, in base al desiderio di rappresentare o meno una vista di “procedura memorizzata centrale” o di una “tabella centrale”⁶.

Il pacchetto “Aree di argomento” contiene pacchetti che raggruppano in modo logico insiemi di tabelle e viste correlati⁷. Si raccomanda di creare le viste nel pacchetto di area di argomento insieme alle tabelle. Tale raccomandazione è solo per motivi organizzativi. Può essere utile avere le viste nell'area argomento dove vengono utilizzate, che le colloca nelle stesse aree di argomento delle tabelle. In questo esempio mostrato in figura 15, ci sono due pacchetti di area di argomento, “Gestione aste” e “Gestione account utente”. Il numero di pacchetti di area di argomento dipende dalla complessità dell'applicazione. Comunque, in generale, i pacchetti di area di argomento del Modello logico di dati “ispirano” i pacchetti di area di argomento del Modello fisico di dati. Le aree di argomento nel Modello logico di dati sono astrazioni delle aree di argomento del Modello fisico di dati.

Le tabelle nei pacchetti di area di argomenti contengono le colonne e i trigger definiti per la tabella. Le tabelle vengono create attraverso una delle seguenti funzioni:

- ☐ funzione di trasformazione XDE da classe a tabella.
- ☐ funzione XDE di reverse engineering di un database esistente.
- ☐ Creazione manuale del progettista di database.

Quando si esegue il reverse engineering di un database esistente, vengono creati uno o più pacchetti di schema nel Modello fisico di dati XDE. I nomi di questi pacchetti si basano sul proprietario di database⁸ del database che viene sottoposto a reverse engineering. Si raccomanda di spostare le tabelle sottoposte a reverse engineering nei pacchetti di area di argomento all'interno del pacchetto “Aree di argomento” e di cancellare i pacchetti di schema sottoposti a reverse engineering. Lo spostamento delle tabelle nei pacchetti di area di argomento organizza le tabelle in modo funzionale per consentire al progettista di database di aggiornarle quando necessario.

I diagrammi che hanno il termine “Vista” nel nome, vengono utilizzati per documentare la Vista dati dell'architettura. Il diagramma “Vista dati: Organizzazione di modello fisico di dati” viene utilizzato per documentare l'organizzazione ad alto livello dei dati del Modello fisico di dati, come espresso nelle principali partizioni (cioè, pacchetti) del Modello fisico di dati. Per ulteriori informazioni sulle viste di architettura, consultare RUP.

8.3 Modello di dominio (facoltativo)

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

⁶

Una vista di tabella centrale consente di comprendere meglio la progettazione/funzionamento del database in un'unica visuale. Una vista centrale di procedura memorizzata semplifica la ricerca e la modifica/mantenimento della procedura memorizzata.

⁷

Alcuni possono mettere in discussione l'uso dei pacchetti di area di argomento nel modello fisico di dati, poiché richiedono manutenzione supplementare per mantenere i pacchetti di area di argomento di database logici e fisici. Le aree di argomento nel modello fisico di dati sono presenti per coerenza con il modello logico di dati (se utilizzato) e ancora di più nel caso dove il modello fisico di dati è “grande” e non c'è alcun modello logico di dati. In questo caso è possibile utilizzare i pacchetti di area di argomento per gestire le tabelle generate dalla trasformazione da classe a tabella.

⁸ Di solito, il database viene sottoposto a reverse engineering una volta e tutti i futuri aggiornamenti vengono sincronizzati utilizzando le funzioni Compare e Sync di XDE.

⁹

All'interno di XDE, il proprietario di database viene acquisito come proprietà del componente <<database>>. Dentro la proprietà Location, come parte della stringa di connessione, è presente un attributo di schema. Quando si esegue il reverse engineering di un database, questo è di solito il proprietario del database.

Nella figura 16 viene mostrato un esempio di struttura raccomandato per il Modello di dominio XDE.¹⁰

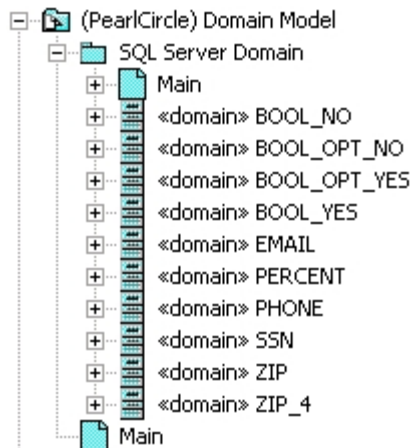


Figura 16: Struttura di modello di dominio XDE

Questo esempio dimostra i valori di dominio di SQL Server organizzati all'interno del pacchetto “Dominio di SQL Server”. Nei casi in cui il progettista di database definisce un grande numero di domini, può essere necessario per il progettista organizzare i domini utilizzando i pacchetti sotto il pacchetto “Dominio di SQL Server”.

9. Modello di implementazione

Il Modello di implementazione, come definito in RUP, contiene gli elementi di implementazione, sia la loro visuale che la rappresentazione fisica (ad esempio, gli elementi UML che rappresentano gli elementi di implementazione, come pure i file fisici nel file system). Il valore di XDE riguardo al Modello di implementazione è la propria capacità di sincronizzare in modo automatico queste rappresentazioni separate tramite ingegnerizzazione roundtrip.

All'interno di XDE, il Modello di implementazione viene rappresentato in più modelli XDE, un esempio dei quali viene mostrato nella figura 17:

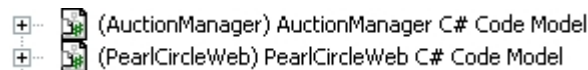


Figura 17: Modelli XDE di implementazione

In questo esempio, sono stati definiti i seguenti file di modello XDE per rappresentare il Modello di implementazione:

- Il “Modello di codice C# AuctionManager” è un file di modello di codice XDE che contiene elementi di codice Visual C# di Microsoft® che costituiscono il Sottosistema di implementazione di Gestore aste. Gli elementi presenti in questo modello partecipano all'ingegnerizzazione roundtrip di XDE.
- Il file di modello XDE “Modello di codice C# PearlCircleWeb” contiene elementi di codice C# di ASP.NET (moduli Web, controlli Web e handler HTTP) che costituiscono di PearlCircle Web il Sottosistema di Implementazione. Tale modello corrisponde a un progetto di applicazione Web di VS.NET. Gli elementi presenti in questo modello partecipano all'ingegnerizzazione roundtrip di XDE.

Tenere presente che può esistere solo un file di modello di codice XDE per progetto di VS.NET. La selezione del numero di progetti e di singoli modelli è una scelta strutturale e può variare per differenti progetti. Per ulteriori informazioni, consultare la guida online di XDE.

Ciascuno di questi modelli viene descritto in modo più dettagliato in una successiva sezione.

¹⁰

All'interno di XDE, vengono supportati diversi database di fornitori, che comprendono DB2, Oracle, Sybase e SQL Server. Quando si crea un Modello di dati XDE di dominio,

il progettista di database creerà tale modello selezionando il database appropriato del fornitore. XDE creerà un elenco predefinito di domini per il database di fornitore selezionato.

9.1 Progetto di libreria di classi per il modello di codice C#

Il “Modello di implementazione di codice C#” contiene elementi che vengono implementati utilizzando il linguaggio C#.

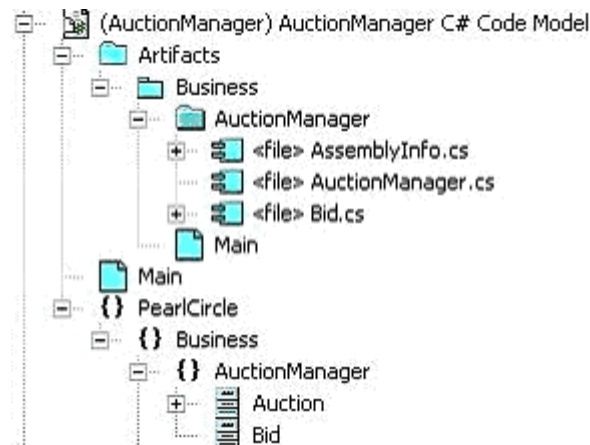


Figura 18: Struttura di "Modello di codice C# di Gestore aste"

In questo esempio, la struttura del “Modello di codice C# di Gestore aste” riflette la struttura del “Modello di progettazione globale” (discusso nella sezione 7). Esiste un pacchetto (che rappresenta uno spazio nomi di .NET) per ogni pacchetto di “Modello di progettazione globale” il cui contenuto verrà implementato in C# (ciò comprende Serviced Components e altre classi C# di supporto). Gli spazi nomi di .NET vengono modellati in XDE come pacchetto con uno stereotipo spazio nomi (compreso tra i simboli '{ }' di parentesi graffe). Poiché il linguaggio di programmazione C# non permette la presenza di spazi nei nomi dello spazio nomi, il nome di uno spazio nomi di C# non può essere identico al nome dell'equivalente elemento di modello di “Modello di progettazione globale”.

Come mostrato in figura 18, il “Modello di codice C#” contiene la rappresentazione visuale dei file di codice sorgente (gli elementi .cs). XDE crea un pacchetto di Artefatti per ogni file di modello di codice che contiene i file di codice sorgente. Tali file di codice sorgente si associano alle classi definite nel “Modello di progettazione globale” (vedere la sezione 7) che sono evolute/maturate al punto in cui possono essere implementate (e nel caso di XDE, essere sottoposte a ingegnerizzazione roundtrip).

Come si può vedere nella figura 18, la struttura di “Modello di codice C#” segue la convenzione di utilizzare il nome dell'azienda come nome iniziale dello spazio nomi di C#. Il nome dell'azienda per l'applicazione di esempio è “Pearl Circle”. Così, i pacchetti contenenti gli elementi di implementazione vengono collocati all'interno dello spazio nomi PearlCircle. Come risultato, tutti gli elementi C# all'interno dello spazio nomi PearlCircle disporranno di un nome completo a cui viene aggiunto il prefisso “PearlCircle”. Ad esempio, il nome completo dello spazio nomi AuctionManager è “PearlCircleBusiness.AuctionManager”. La convenzione di utilizzare il nome dell'azienda come nome iniziale dello spazio nomi di C#, garantisce che i nomi delle classi C# saranno univoci, anche se viene incorporata una libreria di classi C# di una terza parte.

9.2 Modello di codice C# per progetto di applicazione Web

Il modello Web XDE contiene elementi che corrispondono a un progetto di applicazione Web di ASP.NET (Active Server Page .NET). La figura 19 fornisce un esempio del file di modello di codice XDE “PearlCircleWeb” che presenta gli elementi del progetto ASP.NET.

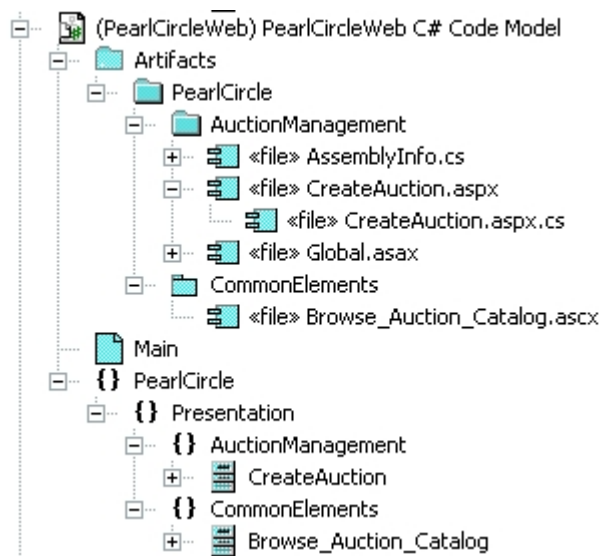


Figura 19: Struttura di "Modello Web C# PearlCircle"

In questo esempio, la struttura del “Modello Web C# PearlCircle” riflette la struttura del “Modello di progettazione globale” (discusso nella sezione 7). È presente uno spazio nomi per ogni pacchetto del “Modello di progettazione globale” il cui contenuto verrà implementato in C# (cioè comprende le classi code-behind e altre classi C# di supporto). Gli spazi nomi di .NET vengono modellati in XDE come pacchetto con uno stereotipo **spazio nomi** (compreso tra i simboli ‘{ }’ di parentesi graffe). Poiché il linguaggio di programmazione C# non permette la presenza di spazi nei nomi dello spazio nomi, il nome di uno spazio nomi di C# non può essere identico al nome dell'equivalente elemento di modello di “Modello di progettazione globale”.¹¹

Le classi code-behind C# (.aspx.cs o .ascx.cs) associate ai moduli Web e ai controlli Web e altre classi C# di supporto come HttpHandlers, possono essere sottoposte a ingegnerizzazione roundtrip tramite XDE. Si noti che attualmente XDE non supporta RTE per i file .aspx o .ascx.¹² La classe **CreateAuction** mostrata in figura 19 modella la classe code-behind che risiede all'interno del file **CreateAuction.aspx.cs**. Nel pacchetto Artefatti, il file **CreateAuction.aspx.cs** viene mostrato sotto il file **CreateAuction.aspx**.

Se tutte le funzionalità strutturalmente significative sono contenute all'interno delle classi code-behind, allora tutto ciò che serve è il modello di codice XDE automaticamente generato. Comunque, nei casi in cui una funzione strutturalmente significativa viene implementata all'interno di un file di controllo Web (.ascx), è possibile aggiungere manualmente al modello una classe corrispondente a tale file. Un esempio di ciò è la classe **Browse_Auction_Catalog** mostrata in figura 19. Si noti che tale classe è stata aggiunta manualmente a questo diagramma.

10. Modello di distribuzione

Il Modello di distribuzione viene rappresentato in un file di “Modello vuoto” XDE denominato “Modello di distribuzione”.

Il “Modello di distribuzione” contiene i nodi e le loro connessioni che rappresentano la configurazione di rete dell'ambiente di distribuzione. Identifica anche gli elementi di implementazione che verranno distribuiti su questi nodi.

Un esempio di “Modello di distribuzione” è mostrato nella figura 20.

¹¹

In Visual Basic.Net, la proprietà di progetto “Spazio nomi Root” deve essere lo spazio nomi che è definito anche per la classe che viene sottoposta a forward engineering. La pratica migliore e consigliata, che si occupa anche di questo problema, è quella di rimuovere il testo dalla proprietà di progetto “Spazio nomi Root” prima di eseguire il reverse engineering del progetto Visual Basic. Tutti gli spazi nomi verranno gestiti in seguito tramite dichiarazioni in codice di spazio nomi.

¹²

I file ASCX e ASPX vengono creati con la forma appropriata e altri tag richiesti. Ad esempio, se si aggiunge un'etichetta o una casella di testo alla classe code-behind XDE, verrà eseguito il forward engineering della variabile membro privata nella classe code-behind. La sola ragione per cui i controlli non vengono visualizzati in Visual Studio Designer è la mancata creazione dei tag in ASPX e del codice in code-behind per creare e posizionare tali elementi. Per ulteriori informazioni, consultare la documentazione di VS.NET.

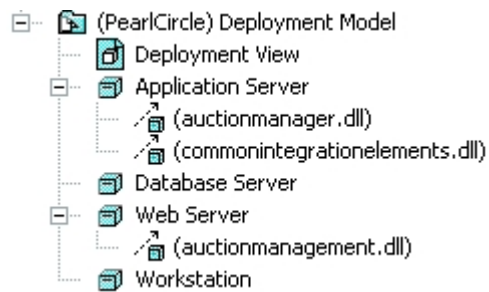


Figura 20: Struttura di modello di distribuzione

In questo esempio, i nodi identificati sono il “Server di database”, il “Server di applicazioni” e il “Server Web”. Il file auctionmanagement.dll viene distribuito sul server Web. I file auctionmanager.dll e commonintegrationelements.dll vengono distribuiti sul server di applicazioni.

Il file di modello “Modello di distribuzione” può contenere anche diagrammi di vista strutturali. Nella figura 20, il diagramma “Vista di distribuzione” rappresenta la vista di distribuzione della struttura. Per ulteriori informazioni sulle viste strutturali, consultare RUP.