

Relazione Progetto Architetture Software Gruppo G8

Francesco Draicchio <fdraicchio@cs.unibo.it>
Andrea Rappini <rappini@cs.unibo.it>
Silvia Righini <righini@cs.unibo.it>
Marco Solieri <solieri@cs.unibo.it>

Relazione Progetto Architetture Software Gruppo G8

di Francesco Draicchio, Andrea Rappini, Silvia Righini, e Marco Solieri

Final version

Data di pubblicazione 14/4/09

Indice

1. Specifica dei requisiti	1
1.1. Tabella riassuntiva dei requisiti	1
1.2. Tabelle dei casi d'uso	2
1.3. Matrice di mappatura dei requisiti	6
2. Relazione di progetto	7
2.1. Modellazione requisiti funzionali	7
2.2. Scelte architetturali	10
2.3. Livello presentazione	12
2.3.1. Scelte progettuali	12
2.3.2. Scelte implementative	12
2.4. Livello Business	22
2.4.1. Introduzione alle scelte progettuali	22
2.4.2. Scelte architetturali	22
2.4.3. Scelte di schieramento	29
2.4.4. Scelte progettuali minori	30
2.5. Livello informativo	32
3. Piano di processo	33
3.1. Obiettivi della gestione di processo	33
3.2. Gestione della pianificazione	33
3.3. Diagrammi di pianificazione	34
3.4. Verifica della pianificazione	43
4. Tool utilizzati	44
4.1. Strumenti per la gestione del progetto	44
4.2. Strumenti per la documentazione	45
4.3. Strumenti per lo sviluppo: application framework, IDE e editor	46
4.4. Strumenti UML	47
4.5. Strumenti per la gestione del processo	48
4.6. Strumenti per il collaudo e la qualità	48
5. Piano di qualità	50
5.1. Portata del piano di qualità	50
5.2. Responsabili delle attività di garanzia della qualità	50
5.3. Pratiche, standard e strumenti a supporto della qualità	50
5.4. Metriche	51
5.5. Attività per la garanzia della qualità	52
5.5.1. Durante tutte le fasi	52
5.5.2. Fase di avvio	52
5.5.3. Fase di elaborazione	53
5.5.4. Fase di costruzione	53
5.5.5. Fase di transizione	54
5.6. Risultati delle misurazioni e dei collaudi sul software	54
6. Manuale utente	57
6.1. Panoramica	57
6.2. Visitatori e clienti	57
6.3. Il carrello	57
6.4. Acquisto	58
7. Valutazione dello sforzo	59
7.1. Valutazione preventiva	59
7.1.1. Calcolo dei punti funzione	59
7.1.2. Stima dello sforzo	60
7.2. Valutazione consuntiva	60
8. Diari	62
8.1. Diario di Gruppo	62
8.2. Diario di Marco Solieri	64
8.3. Diario di Andrea Rappini	67
8.4. Diario di Silvia Righini	69

Relazione Progetto Architetture
Software Gruppo G8

8.5. Diario di Francesco Draicchio	72
Glossario	74
Bibliography	75

Lista delle figure

2.1. Diagramma dei casi d'uso	7
2.2. Diagramma dell'attività complessiva	8
2.3. Diagramma dell'attività Cassa	8
2.4. Diagramma degli stati relativi all'attività Cassa	9
2.5. Deployment diagram del progetto G8Bookshop	11
2.6. Panoramica su package e cartelle	13
2.7. g8.bookshop.presentation.servlet	14
2.8. g8.bookshop.presentation.Constants	15
2.9. WebContent: contenitore per pagine JSP e JSPF	16
2.10. Web Content: relazioni tra pagine JSP e frammenti JSPF	16
2.11. g8.bookshop.presentation.content: gestione dei contenuti	17
2.12. g8.bookshop.presentation: relazioni tra classi e package del progetto g8Presentation	18
2.13. g8.bookshop.business.ws.catalogueservice: un esempio di web service	19
2.14. g8.bookshop.business.ws: diagramma completo dei web service	19
2.15. Diagramma di sequenza per il caso d'uso Autenticazione	20
2.16. g8.bookshop.business	23
2.17. g8.bookshop.business.ws	24
2.18. g8.bookshop.business.um	25
2.19. g8.bookshop.business.core	26
2.20. g8.bookshop.business.persistence	27
2.21. g8.bookshop.business.util	28
2.22. g8.bookshop.business	29
2.23. Diagramma di sequenza	30
3.1. Diagramma di Gant preventivo (1 di 4)	35
3.2. Diagramma di Gant preventivo (2 di 4)	36
3.3. Diagramma di Gant preventivo (3 di 4)	37
3.4. Diagramma di Gant preventivo (4 di 4)	38
3.5. Diagramma di Gant consuntivo (1 di 4)	39
3.6. Diagramma di Gant consuntivo (2 di 4)	40
3.7. Diagramma di Gant consuntivo (3 di 4)	41
3.8. Diagramma di Gant consuntivo (4 di 4)	42
5.1. Metriche sulle classi	54
5.2. Metriche sui metodi	55
5.3. Metriche sulle classi	56

Lista delle tabelle

1.1. UC01 (RicercaLibri)	2
1.2. UC02 (SelezioneLibri)	3
1.3. UC03 (Autenticazione)	4
1.4. UC04 (Acquisto)	4
1.5. UC05 (Disconnessione)	5
1.6. UC06 (ModificaCarrello)	5
1.7. UC07 (VisualizzaCarrello)	6
7.1. Calcolo punti funzione non aggiustati	59
7.2. Valutazione consuntiva dello sforzo	60
8.1. Diario del Project Manager	64
8.2. Diario del Quality Engineer	67

Capitolo 1. Specifica dei requisiti

1.1. Tabella riassuntiva dei requisiti

Identificativo	Priorità	Classificazione	Descrizione
R1F.1	1	Funzionale	Ricerca nel catalogo per autore e per titolo
R1F.2	1	Funzionale	Selezione libri per l'acquisto da parte di utenti autenticati (inserimento nel carrello)
R1F.3	1	Funzionale	Acquisto del carrello da parte di utenti autenticati
R1N	1	Non funzionale	Sviluppo su piattaforma JBOSS 5.0
R2N	2	Non funzionale	Utilizzo della versione 3.0 della tecnologia EJB
R3N.1	3	Non funzionale	Schieramento dell'applicazione in cluster
R3N.2	3	Non funzionale	Comunicazione fra interfaccia utente e livello business mediante web service.
R4F	4	Funzionale	Autenticazione utente
R4N.1	4	Non funzionale	Memorizzazione persistente del catalogo in memoria secondaria
R4N.2	4	Non funzionale	Realizzazione autenticazione con validazione di nome utente e password
R4N.3	4	Non funzionale	Realizzazione interfaccia utente con pagine web
R5N.1	5	Non funzionale	Bilanciamento di carico mediante un bilanciatore esterno realizzato con Apache HTTP Server 2.x
R5N.2	5	Non funzionale	Realizzazione della persistenza mediante un database MySql

1.2. Tabelle dei casi d'uso

Tabella 1.1. UC01 (RicercaLibri)

Caso d'uso: RicercaLibri
ID: UC01
Attori coinvolti. Utente
Funzione. Permettere la ricerca di libri secondo uno o più parametri.
Precondizioni. Nessuna.
Trigger (innesco). L'utente desidera cercare uno o più libri.
Sequenza degli eventi.
<ol style="list-style-type: none"> 1. L'utente inserisce i valori sui quali effettuare la ricerca nei campi della maschera di ricerca e conferma. 2. Il sistema restituisce l'elenco di tutti i libri del catalogo coerenti coi valori di ricerca immessi. Nel caso nessun libro risponda ai criteri di ricerca, il sistema lo notifica con un messaggio appropriato.
Eccezioni.
<ol style="list-style-type: none"> 1. Il sistema incontra un errore durante l'accesso al catalogo e la ricerca fallisce: in questo caso l'utente viene notificato con un messaggio appropriato e gli viene suggerita un'azione alternativa(es. riprovare più tardi.)
Postcondizioni.

Tabella 1.2. UC02 (SelezioneLibri)

Caso d'uso: SelezioneLibri
ID: UC02
Attori coinvolti. Cliente
Funzione. Permettere al cliente di selezionare uno o più libri per un eventuale successivo acquisto.
Precondizioni. Il cliente deve aver effettuato una ricerca (caso d'uso UC01) e trovarsi di fronte all'elenco di libri (non vuoto) mostratogli dal sistema.
Trigger (innesco). Il cliente autenticato desidera selezionare alcuni libri di suo interesse per acquistarli.
Sequenza degli eventi.
<ol style="list-style-type: none"> 1. include(RicercaLibri) 2. il cliente indica uno o più libri (eventualmente specificandone la quantità) attraverso l'interfaccia, che deve presentare questa possibilità in modo chiaro per ogni singolo libro. 3. Il sistema inserisce i libri, nella quantità specificata, nel carrello del cliente.
Sequenza alternativa.
<ol style="list-style-type: none"> 1. Il cliente non desidera selezionare nessuno dei libri risultanti dalla sua ricerca. Abbandona la schermata ed effettua una nuova ricerca, passando nuovamente al caso d'uso RicercaLibri.
Postcondizioni. I libri selezionati sono inseriti all'interno del carrello del cliente, in aggiunta ad eventuali libri già presenti. Qualora il cliente avesse selezionato libri già inseriti nel carrello in precedenza, il numero di nuove copie viene sommato alle copie già inserite. Il carrello non è vuoto ed è quindi possibile acquistarne il contenuto.
Note.
<ol style="list-style-type: none"> 1. La conferma dell'operazione può avvenire o meno, dipende dall'interfaccia.

Tabella 1.3. UC03 (Autenticazione)

Caso d'uso: Autenticazione
ID: UC03
Attori coinvolti.
<ul style="list-style-type: none"> • Visitatore • Cliente
Funzione. Permettere ad un visitatore di autenticarsi presso il sistema e al fine di accedere alle funzioni di selezione e acquisto libri.
Precondizioni. Il visitatore deve possedere un login e una password.
Trigger (innesco). Il visitatore desidera acquistare dei libri nell'immediato o sceglierne alcuni per un acquisto in un secondo momento.
Sequenza degli eventi.
<ol style="list-style-type: none"> 1. Il visitatore identifica nell'interfaccia la sezione dedicata all'autenticazione. 2. Il visitatore inserisce il suo username e la sua password. 3. Se le credenziali del visitatore risultano corrette (il login corrisponde a un login riconosciuto e la password inserita è quella attesa per il login dato) <ol style="list-style-type: none"> a. Il sistema identifica il visitatore e cambia il suo stato in cliente. <p>Altrimenti, se il login non corrisponde a un login valido o la password non corrisponde al login</p> <ol style="list-style-type: none"> b. Il sistema notifica il visitatore con un messaggio di errore generico di autenticazione fallita. c. Il visitatore può riprovare l'autenticazione ripartendo al punto 2.
Postcondizioni. L'utente è riconosciuto dal sistema e gli è possibile selezionare libri e procedere all'acquisto. L'utente passa quindi dallo stato di Visitatore a quello di Cliente.

Tabella 1.4. UC04 (Acquisto)

Caso d'uso: Acquisto
ID: UC04
Attori coinvolti. Cliente
Funzione. Permette di acquistare l'intero contenuto del carrello.
Precondizioni. Il carrello non è vuoto.
Trigger (innesco). Il cliente desidera acquistare il contenuto del suo carrello.
Sequenza degli eventi.
<ol style="list-style-type: none"> 1. include(SelezionaLibri) 2. include(VisualizzaCarrello) 3. extends(ModificaCarrello): il cliente può entrare nel caso d'uso ModificaCarrello. 4. Il cliente specifica al sistema mediante l'interfaccia preposta la volontà di acquistare i libri nel suo carrello. 5. Il sistema conferma l'acquisto.
Postcondizioni. Il carrello è svuotato.

Tabella 1.5. UC05 (Disconnessione)

Caso d'uso: Disconnessione
ID: UC05
Attori coinvolti. Cliente
Funzione. Permette di terminare la comunicazione col sistema da parte di un cliente.
Precondizioni. Nessuna.
Trigger (innesco).
<ul style="list-style-type: none"> • Possibile innesco 1: il cliente desidera non essere più riconosciuto dal sistema (tipicamente perché ha terminato di utilizzarlo o non desidera più fare acquisti). • Possibile innesco 2: il sistema al termine di un acquisto ritiene conclusa la comunicazione con un cliente e lo disconnette dal sistema.
Sequenza degli eventi.
<ol style="list-style-type: none"> 1. Il cliente o il sistema richiedono la disconnessione del cliente. 2. Il cliente viene disconnesso dal sistema e ritorna nello stato di Visitatore.
Postcondizioni. Nessuna.

Tabella 1.6. UC06 (ModificaCarrello)

Caso d'uso: ModificaCarrello
ID: UC06
Attori coinvolti. Cliente
Funzione. Permette di modificare il numero di copie specificato per ogni libro presente nel carrello, fino a portarlo a zero (che coincide con l'eliminazione del libro dal carrello.)
Precondizioni. Il cliente ha inserito almeno un libro nel carrello. L'utente sta visualizzando il carrello.
Trigger (innesco). L'utente desidera modificare il contenuto del suo carrello eliminando libri o modificando per un libro inserito il numero di copie.
Sequenza degli eventi.
<ol style="list-style-type: none"> 1. L'utente modifica il valore indicante la quantità per uno o più libri 2. L'utente conferma la modifica. 3. Il sistema mostra il nuovo carrello con le quantità modificate. I libri la cui quantità è stata portata a zero o a un numero negativo vengono eliminati e non compaiono più nel carrello.
Postcondizioni. Il nuovo carrello è visualizzato. I libri contenuti nel carrello hanno subito modifiche.

Tabella 1.7. UC07 (VisualizzaCarrello)

Caso d'uso: VisualizzaCarrello
ID: UC07
Attori coinvolti. Cliente
Funzione. Mostrare il contenuto del carrello del cliente.
Precondizioni. Nessuna.
Trigger (innesco). Il cliente desidera visualizzare i libri che ha selezionato.
Sequenza degli eventi.
1. Il cliente richiede di vedere il contenuto del suo carrello.
2. Il sistema presenta al cliente la lista dei libri che egli ha inserito nel carrello (in vari momenti o anche in varie sessioni distinte). Qualora nessun libro sia presente nel carrello il sistema notifica che il carrello è vuoto.
Postcondizioni. Nessuna.

1.3. Matrice di mappatura dei requisiti

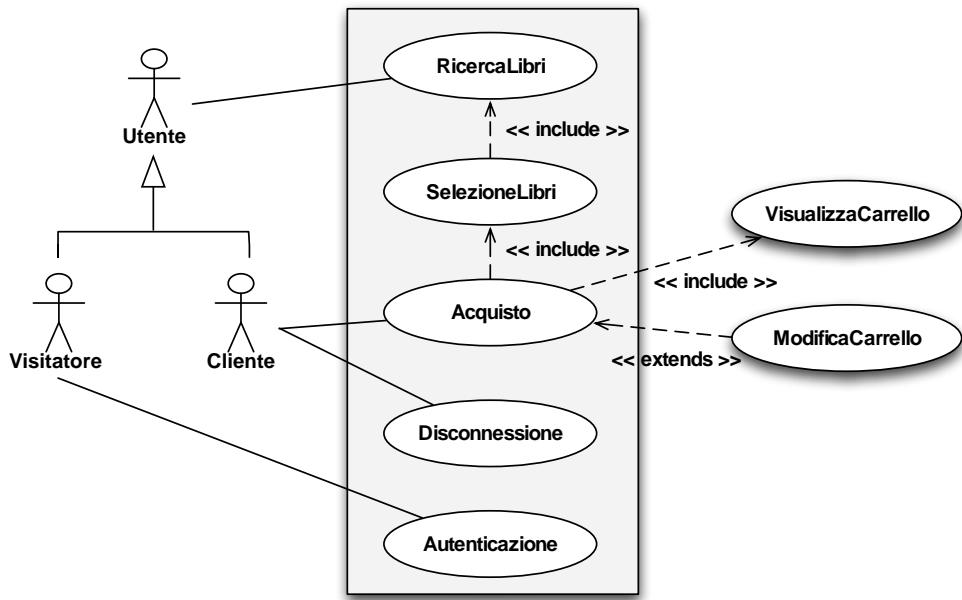
	UC01	UC02	UC03	UC04	UC05
R1F.1	X				
R1F.2		X			
R1F.3				X	
R4F			X		

Capitolo 2. Relazione di progetto

2.1. Modellazione requisiti funzionali

A partire dai requisiti funzionali si è partiti nell'analizzare ciò che il sistema doveva offrire ai suoi utenti. Il diagramma dei casi d'uso in Figura 2.1, «Diagramma dei casi d'uso.» modella le possibili azioni per un utente.

Figura 2.1. Diagramma dei casi d'uso.



Ogni utente può essere un visitatore, ovvero un utente non autenticato e di conseguenza non riconosciuto dal sistema, o un cliente. Il solo caso d'uso comune ai due attori è la Ricerca (UC01), anche se nel caso del cliente questa rientra in un caso d'uso più ampio detto Acquisto (UC04).

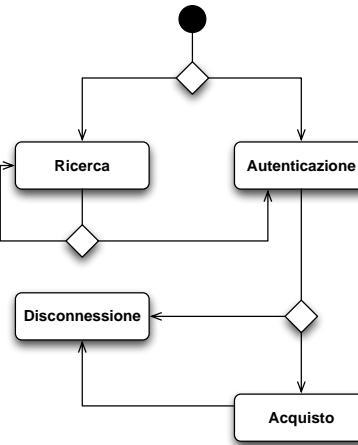
Ogni utente inizia la sua interazione col sistema come visitatore: oltre alla ricerca egli può quindi effettuare l'Autenticazione, che modifica il suo stato e lo trasforma in un cliente. Un cliente, a sua volta, può entrare nel caso d'uso Disconnessione (UC05) e riprendere il ruolo di visitatore. Il visitatore può solo autenticarsi (Autenticazione, UC03) o effettuare ricerche (Ricerca, UC01). Il cliente può invece entrare nel caso d'uso Acquisto (UC04). Il caso d'uso Acquisto include prima di tutto altri due casi d'uso: Ricerca (UC01) e Selezione (UC02). Infatti il caso d'uso Acquisto inizia in modo analogo al caso d'uso Ricerca per il visitatore, e terminata la Ricerca entra nel caso d'uso Selezione. Terminate le operazioni di questi due casi d'uso, l'Acquisto include anche il caso d'uso VisualizzaCarrello (UC06), poiché la visualizzazione è necessaria prima della conferma dell'acquisto. Il cliente può anche decidere di attivare il caso d'uso ModificaCarrello (UC07), prima di svolgere i passi del caso d'uso Acquisto (UC04) vero e proprio, che terminano l'acquisto. Le schede dei casi d'uso (dei quali viene riportato l'ID) possono essere riviste nella sezione 1.2.

La modellazione precedente tuttavia illustra le funzionalità del sistema offerte all'utente ma non descrive in modo adeguato l'ordine delle diverse azioni. Mostriamo quindi la sequenza delle attività con diagrammi più adeguati.

Il diagramma delle attività in Figura 2.2, «Diagramma dell'attività complessiva.» rappresenta in modo generale le diverse attività. All'arrivo di un visitatore su G8Bookshop, egli può effettuare la ricerca o l'autenticazione. Se sceglie di effettuare una ricerca, al termine di essa può ripetere la attività effettuando nuove ricerche, o passare all'attività autenticazione. Scegliendo invece di autenticarsi, passerà a un

ramo diverso del diagramma (a cui appunto si accede solo passando attraverso l'autenticazione) nel quale gli viene fornita la possibilità di effettuare acquisti. Una volta entrato in questo ramo egli può in ogni momento interrompere l'attività corrente e disconnettersi: in questo modo ritorna al punto di partenza del diagramma (aspetto non modellato per semplicità). L'attività Acquisto può essere ripetuta a piacere.

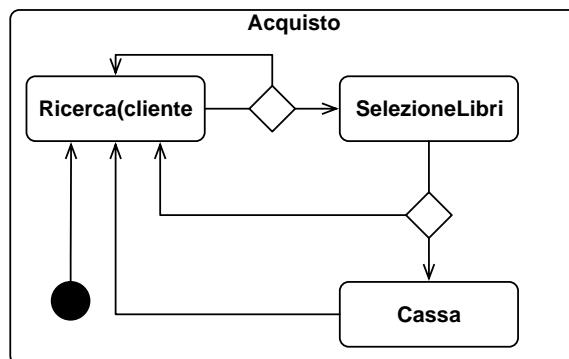
Figura 2.2. Diagramma dell'attività complessiva.



Osserviamo più nel dettaglio l'attività Acquisto: essa è una macro attività composta di tre sottoattività. Appena iniziata l'attività Acquisto, il cliente si troverà a svolgere l'attività Ricerca; una volta effettuata una ricerca egli può effettuarne di nuove o procedere alla selezione di alcuni dei libri risultanti. Dopo una selezione, può tornare a una nuova ricerca (o più di una) oppure procedere all'attività Cassa, nella quale potrà acquistare i libri selezionati.

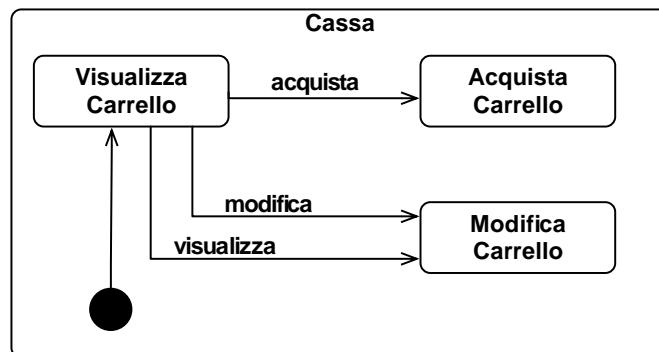
Dopo aver effettuato l'attività Cassa, e anche durante, è possibile tornare all'attività Ricerca. Ricordiamo che in ogni momento è possibile abbandonare qualsiasi attività di Acquisto iniziando l'attività Disconnessione.

Figura 2.3. Diagramma dell'attività Cassa.



Il diagramma in figura Figura 2.4, «Diagramma degli stati relativi all'attività Cassa.» riporta gli stati possibili relativi all'attività Cassa. Appena iniziata l'attività Cassa, il Cliente si trova nello stato VisualizzaCarrello. Egli può notificare la sua volontà di acquistare il contenuto del suo carrello, passando così allo stato finale AcquistaCarrello (ed esce di conseguenza dall'attività Cassa). Alternativamente, egli può richiedere la modifica del suo carrello passando allo stato ModificaCarrello: da qui può poi tornare allo stato VisualizzaCarrello e riprendere da qui. In ogni momento è possibile uscire da qualsiasi stato di Cassa muovendosi verso l'attività Ricerca o Disconnessione.

Figura 2.4. Diagramma degli stati relativi all'attività Cassa.



2.2. Scelte architetturali

J2EE offre la possibilità di realizzare applicazioni web basate su un modello architettonico a 4 livelli, ovvero Client tier, Web tier, Business tier e Information tier. Tale caratteristica stimola in particolar modo a progettare i componenti software che costituiscono l'applicazione in maniera fortemente modulare, forzando così la flessibilità e propendendo naturalmente ad una logica di riutilizzo dei componenti software.

Tipicamente però, è comune pratica accoppiare i due livelli centrali del modello in un unico tier. In questo modo si ottiene un forte accoppiamento tra il livello web e quello business e non si sfrutta pienamente il paradigma proposto dalla tecnologia in questione.

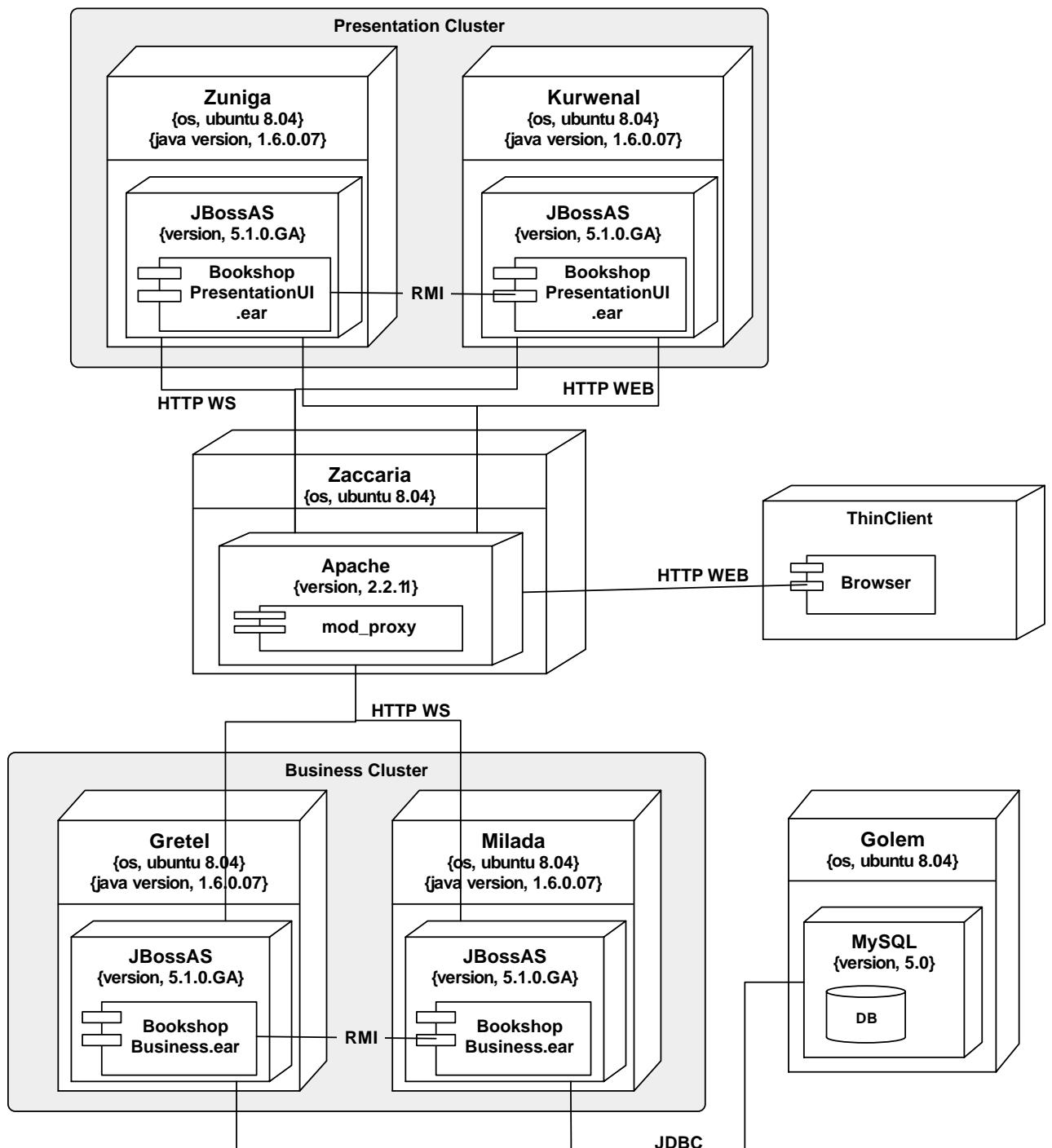
Quindi, considerando anche che le specifiche del progetto impongono l'utilizzo di Web Service, e quindi di uno standard, per realizzare la comunicazione tra il livello business e quello web, abbiamo ritenuto ragionevole adottare strettamente il modello architettonico J2EE, e quindi separare in maniera netta la logica di presentazione da quella business.

Infatti una peculiarità del nostro modello è appunto la "traduzione" da Web tier a Presentation tier. Tale distinzione ci consente di stabilire in maniera ottimale le responsabilità di ogni componente e quindi di non sconfinare con processi business nella logica di presentazione, mantenendo così il livello di accoppiamento tra i due tier molto basso. Questa scelta è stata fatta anche in base all'assunzione che, in un'ottica di possibili sviluppi futuri, gioca un importante ruolo la flessibilità, in caso di cambiamenti sostanziali sia all'interfaccia utente sia al core dell'applicazione.

Per quanto riguarda lo schieramento, è stata una scelta quasi obbligata separare presentazione e business in due differenti cluster, questo per l'evidente e ricercata distinzione logica tra i componenti e per necessità riguardanti le differenti metodologie di replicazione adottate per i dati di sessione. Entrambi cluster sono infatti statefull.

Infine, per quanto riguarda il bilanciamento di carico, entrambi i cluster vengono gestiti da un unico server. In caso di collo di bottiglia, la replicazione di quest'ultimo con le opportune configurazioni non rappresentano assolutamente un ostacolo. Il diagramma in Figura 2.5, «Deployment diagram del progetto G8Bookshop» illustra la struttura del progetto G8Bookshop schierato.

Figura 2.5. Deployment diagram del progetto G8Bookshop



2.3. Livello presentazione

2.3.1. Scelte progettuali

Logica di presentazione

La logica di presentazione è stata implementata, in puro stile Java Enterprise, utilizzando sia servlet che JSP, in modo da dividere quanto più possibile la parte puramente presentazionale, ovvero il codice xhtml, dalla logica di interazione. Questa scelta di utilizzare entrambe le tecnologie è il solo modo nel quale ci è sembrato possibile mantenere il codice pulito da ogni traccia di markup, obiettivo a cui tenevamo molto. Le JSP, d'altro canto, sono state create modulari, composte di frammenti JSP (JSPF) rappresentanti elementi ripetuti e significativi, come la parte di head del documento, il titolo, il piè di pagina e i menu. Questa suddivisione permette naturalmente semplici e veloci modifiche al layout, senza contare che naturalmente gran parte dello stile è a parte, descritto in un CSS. Per la comunicazione tra servlet e JSP e viceversa, si è utilizzato un POJO condiviso, nel quale ad esempio la servlet è solita inserire i risultati di ritorno delle chiamate ai web service, prima di trasferire il controllo alla JSP. La JSP a quel punto può comodamente leggere le informazioni dall'oggetto condiviso e inserirle adeguatamente formattate, nel punto corretto del codice XHTML. La formattazione dei risultati XML in arrivo dal web service viene svolta all'interno dei metodi getter del POJO, attraverso una trasformazione XSLT.

Caching dei risultati delle ricerche

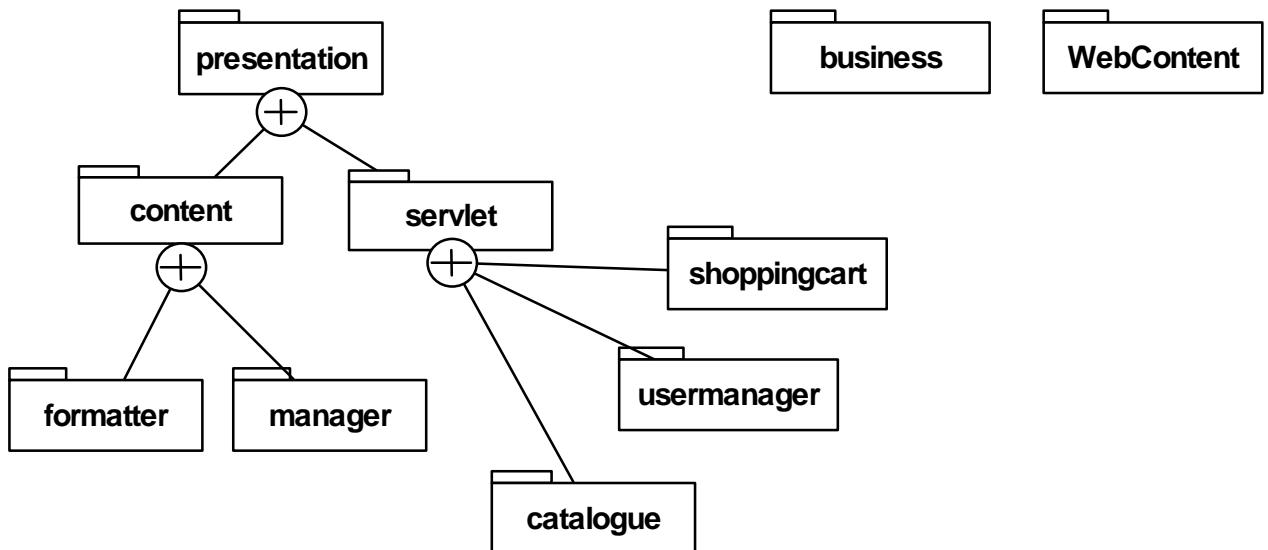
Durante il processo di navigazione, l'utente sottopone le servlet ad alcuni passaggi di gestione. Per la logica di presentazione alcuni di questi "passaggi" richiedono visualizzazioni differenti dei risultati di ricerca. Questa condizione si presenta nel passaggio tra ricerca semplice e ricerca autenticata, che consente la selezione dei libri mediante un'apposita interfaccia. Con un occhio di riguardo alle performance e grazie alla natura del formato dei dati che vengono scambiati tramite web service tra i livelli di presentazione e di business è stato possibile realizzare un meccanismo di caching dei risultati delle ricerche dell'utente, basato sull'utilizzo delle sessioni http, che consente di non effettuare ulteriori invocazione ai servizi di ricerca lato business ogni qualvolta si presenta la necessità di avere due rappresentazioni differenti dello stesso risultato. Per entrare brevemente nei dettagli implementativi, tale semplice operazione si realizza memorizzando in variabili di sessione il risultato dell'ultima ricerca in formato XML, utilizzando, a seconda delle necessità, differenti trasformazioni XSLT per le rispettive visualizzazioni.

Replicazione HTTP

Per tollerare guasti di tipo crash tra i nodi del cluster di presentazione e garantire la continuità del servizio, è stato deciso di attivare il supporto per la replicazione della sessione http che offre JBoss. In relazione a quanto accennato nella sezione precedente riguardo alla netta separazione logica dei due livelli centrali, è doveroso specificare che in tali sessioni non si trovano informazioni o dati utili alla gestione di procedure legate allo stato della sessione che riguardino il lato business, come ad esempio lo stato del carrello, ma viene solamente tenuta traccia dell'identificativo della sessione con il quale l'infrastruttura di gestione degli utenti compirà ogni aggiornamento di stato lato business.

2.3.2. Scelte implementative

In questa sezione verrà illustrata la struttura del lato presentation del progetto Bookshop. Il diagramma in Figura 2.6, «Panoramica su package e cartelle» descrive la struttura complessiva del progetto, organizzato in package. I diversi package verranno esposti più in dettaglio nelle sezioni che seguiranno.

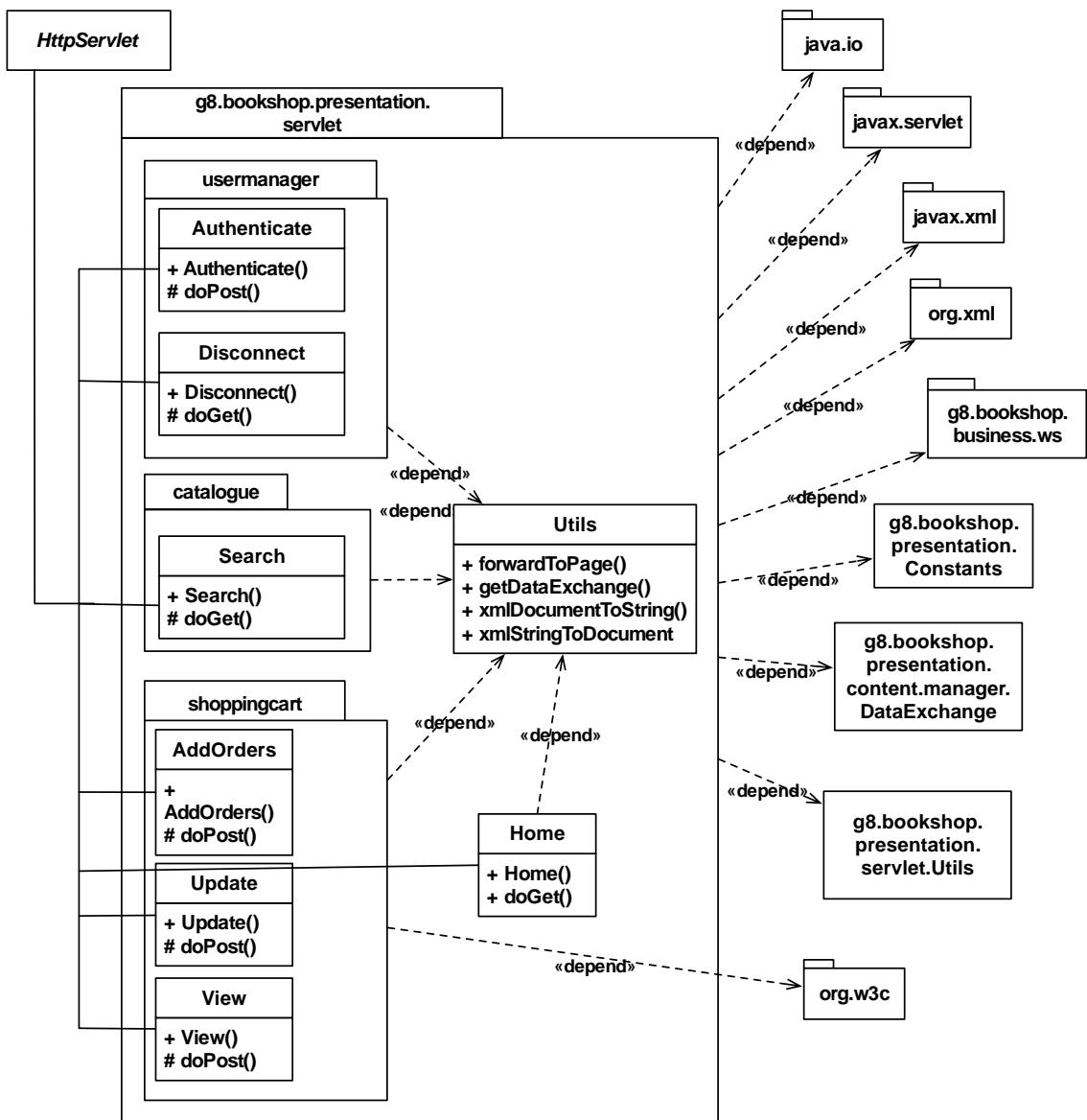
Figura 2.6. Panoramica su package e cartelle

La parte di presentazione è raccolta in un progetto dal nome `g8.bookshop.presentation`. Esso contiene due macro package: `business` e `presentation`. Il primo, `business`, contiene le classi che permettono alle servlet di connettersi ai web service sottostanti. Il secondo, più complesso, contiene la parte di presentazione vera e propria, divisa nelle due parti `servlet` e `content`. Il package `servlet` naturalmente raggruppa le classi che implementano l'interfaccia `javax.servlet.http.HttpServlet`, mentre il package `content` raccoglie classi e altri file relativi alla gestione del contenuto. Spiegazioni più accurate sul ruolo dei package descritti e sulle loro classi verranno fatti in seguito.

Essenziale nella presentazione è anche la cartella `WebContent`, la quale contiene le pagine JSP del progetto, i frammenti JSPF usati per modularizzare le pagine stesse, e i fogli di stile CSS.

Servlet

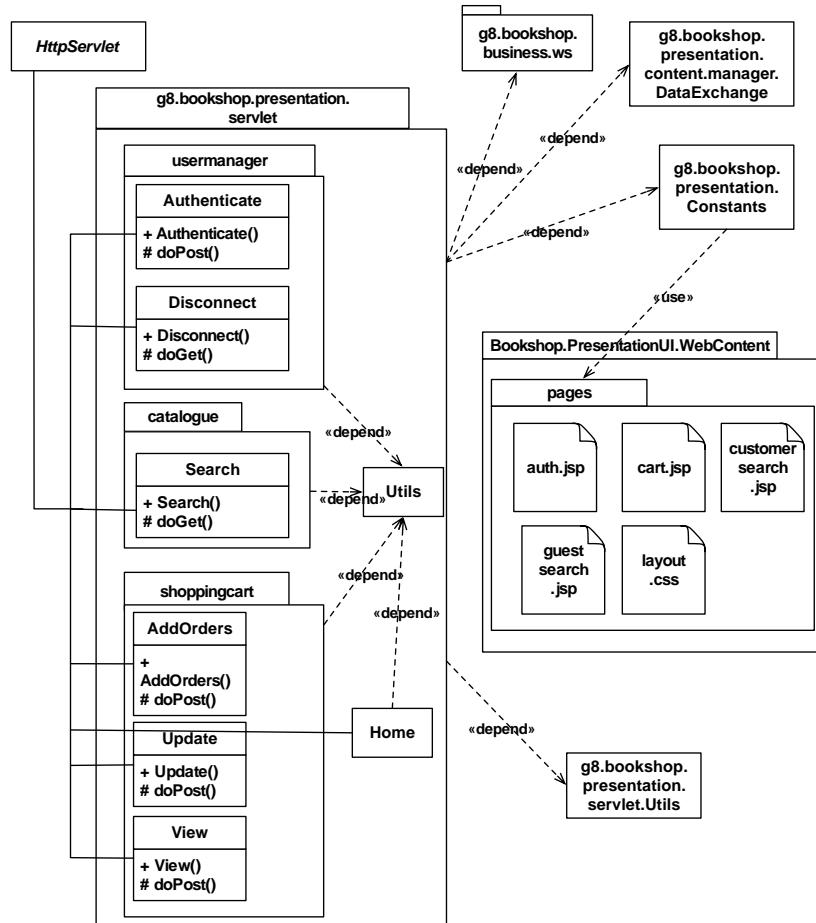
Figura 2.7. g8.bookshop.presentation.servlet



Il pacchetto `g8.bookshop.presentation.servlet` raggruppa le classi che implementano l'interfaccia `HttpServlet`. Contiene tre pacchetti, `userManager`, `catalogue` e `shoppingcart` i quali contengono le servlet che si occupano, nell'ordine, di autenticazione e disconnessione, della gestione del catalogo (la ricerca), e della gestione del carrello (visualizzazione e modifica). La servlet `Home` si trova nel package principale, insieme al file `Utils.java` che fornisce metodi ausiliari per la manipolazione di stringhe XML, per il passaggio di controllo da una servlet a una JSP e per la gestione dell'oggetto condiviso tra servlet e JSP che permette il passaggio di dati tra le due.

Il trasferimento del controllo tra servlet e JSP è mediato dalla classe `g8.bookshop.presentation.Constants`, la quale contiene, oltre a diverse costanti del progetto, ogni riferimento ai file JSP. Questo collegamento è illustrato nel diagramma in Figura 2.8, «`g8.bookshop.presentation.Constants`».

A titolo di esempio, in questo diagramma sono state rappresentate anche le librerie esterne utilizzate dal package.

Figura 2.8. g8.bookshop.presentation.Constants

WebContent: pagine JSP

La cartella WebContent contiene le JSP e i frammenti JSPF. In Figura 2.9, «WebContent: contenitore per pagine JSP e JSPF» sono rappresentati i file contenuti. In Figura 2.10, «Web Content: relazioni tra pagine JSP e frammenti JSPF» si sintetizza come i frammenti JSPF sono inclusi nelle diverse pagine JSP.

Figura 2.9. WebContent: contenitore per pagine JSP e JSF

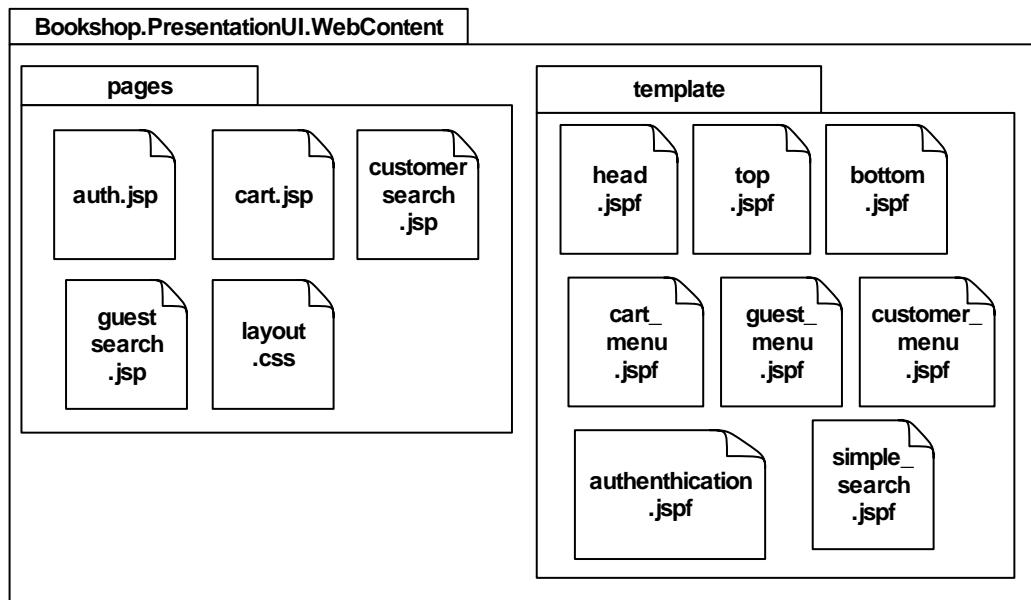
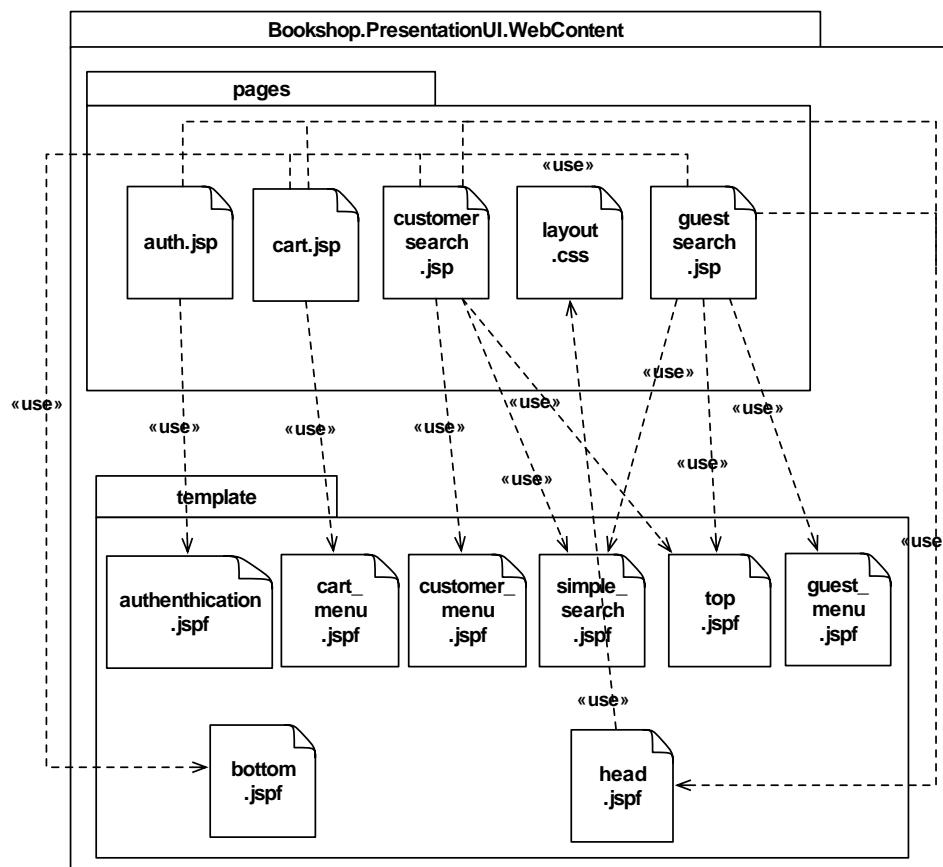


Figura 2.10. Web Content: relazioni tra pagine JSP e frammenti JSF

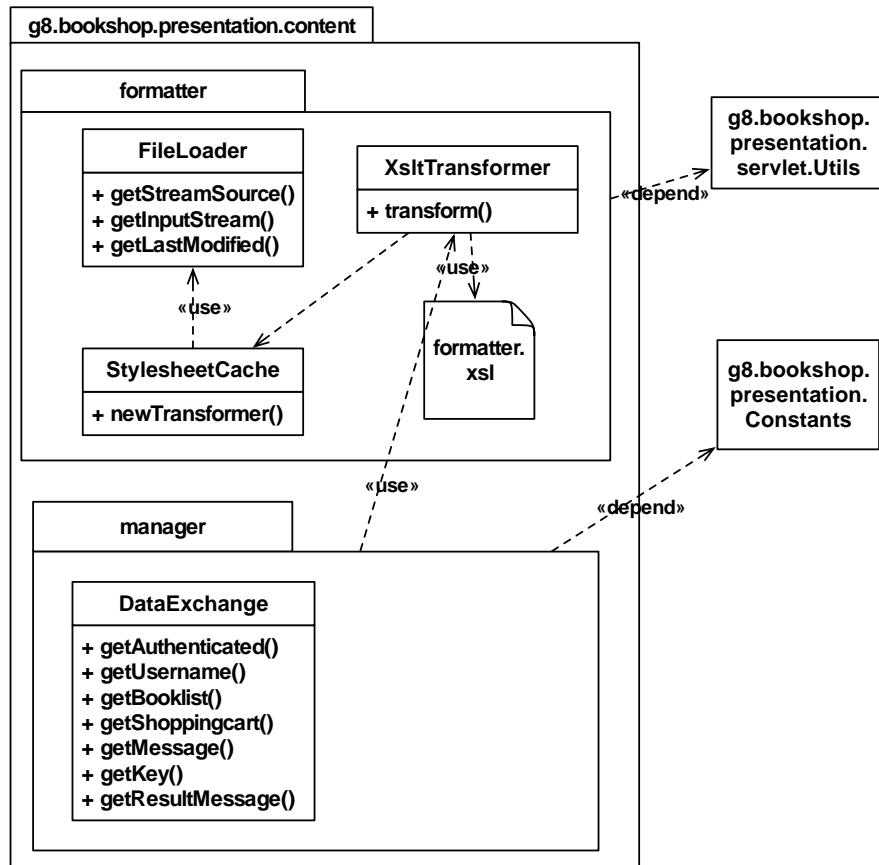


Content: gestione, manipolazione e formattazione dei contenuti

Il diagramma in Figura 2.11, «g8.bookshop.presentation.content: gestione dei contenuti.» rappresenta il package delegato alla gestione dei contenuti, con le classi per la trasformazione XSLT del XML

proveniente dal lato business, e la classe dedicata alla condivisione dei contenuti tra JSP e servlet. Vengono poi illustrate in Figura 2.12, «g8.bookshop.presentation: relazioni tra classi e package del progetto g8Presentation» le relazioni tra i diversi package che compongono l'intero lato presentazione.

Figura 2.11. g8.bookshop.presentation.content: gestione dei contenuti.

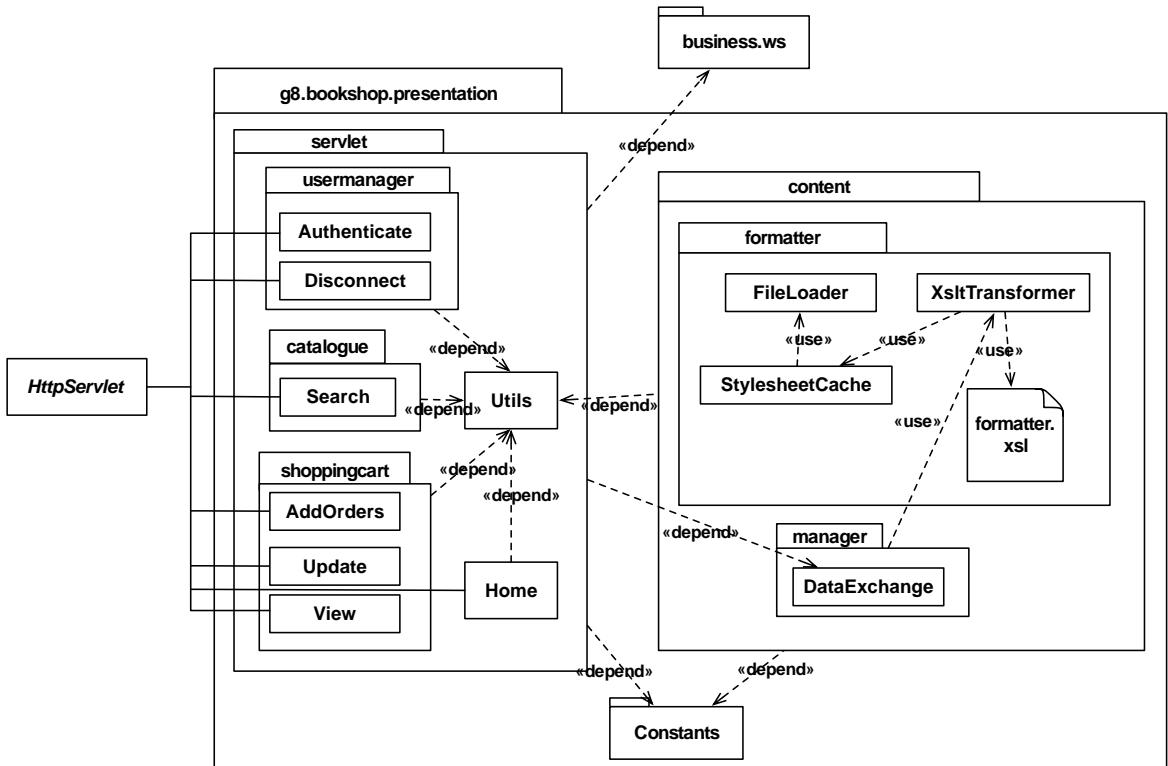


Il package manager contiene una sola classe, `DataExchange`, che viene utilizzata come oggetto condiviso tra servlet e JSP per il passaggio di informazioni e contenuti tra le due; essa contiene esclusivamente metodi getter e setter (tralasciati per semplicità nel diagramma).

Il package `formatter` si occupa invece dell'interpretazione dei dati che le servlet ricevono dal lato business: le liste di libri risultanti da una ricerca, o il contenuto di un carrello. La classe `XsltFormatter` è la classe centrale del package. Essa utilizza un foglio di stile XSLT per la trasformazione dell'XML ricevuto in XHTML pronto per la visualizzazione. Per farlo, carica il foglio di stile attraverso la classe `StylesheetCache` che a sua volta si avvale dei metodi forniti dalla classe `FileLoader`. `StylesheetCache` è una classe singleton che implementa una cache per i fogli di stile in modo da evitare di ricaricare più volte lo stesso foglio di stile ad ogni invocazione del trasformatore.

Il diagramma in Figura 2.12, «g8.bookshop.presentation: relazioni tra classi e package del progetto g8Presentation» rappresenta le relazioni

Figura 2.12. g8.bookshop.presentation: relazioni tra classi e package del progetto g8Presentation



In questo diagramma è rappresentato l'intero lato presentation con le relazioni tra le sue classi e i suoi package. Praticamente l'intero lato presentation dipende dalle classi di ausilio `Utils` e `Constants`. Le diverse servlet sono indipendenti tra loro, mentre utilizzano le classi del package `business.ws` per la connessione ai web service, le informazioni nella classe `Constants` per richiamare le JSP, e la classe `DataExchange` per lo scambio di informazioni con le JSP.

Quest'ultima classe, `DataExchange` è la sola ad utilizzare i metodi della classe `XsltTransformer`, la quale a sua volta è la sola ad utilizzare i metodi delle altre classi del suo package.

Web service

In questa sezione due diagrammi descrivono i web service utilizzati nel progetto. Alcune di queste classi sono state generate automaticamente a partire dalle altre. Non tutte vengono utilizzate.

Nel diagramma in Figura 2.13, «`g8.bookshop.business.ws.catalogueservice`: un esempio di web service» è rappresentato uno dei tre sottoinsiemi di queste classi, ovvero tutte le classi relative al `CatalogueService`. Tutte le classi del package sono poi riassunte nel diagramma in Figura 2.14, «`g8.bookshop.business.ws`: diagramma completo dei web service».

Figura 2.13. g8.bookshop.business.ws.catalogueservice: un esempio di web service

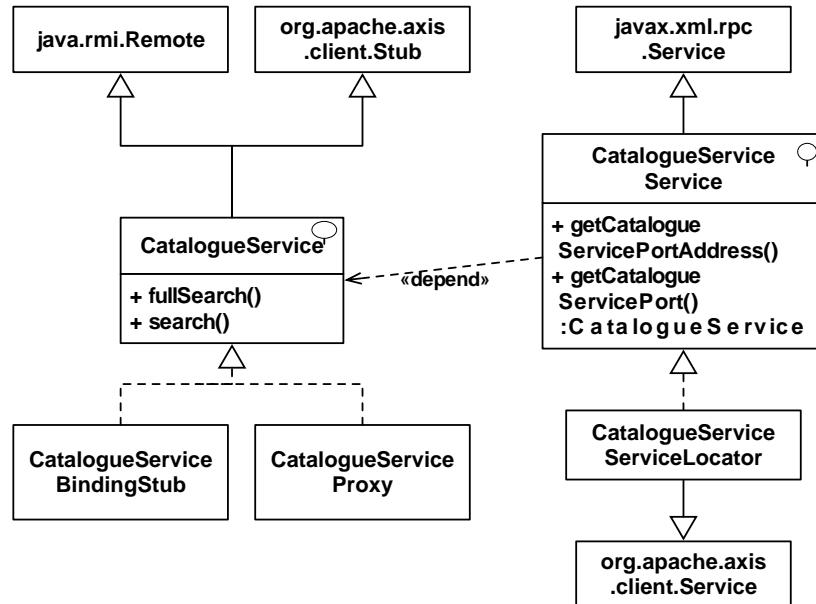
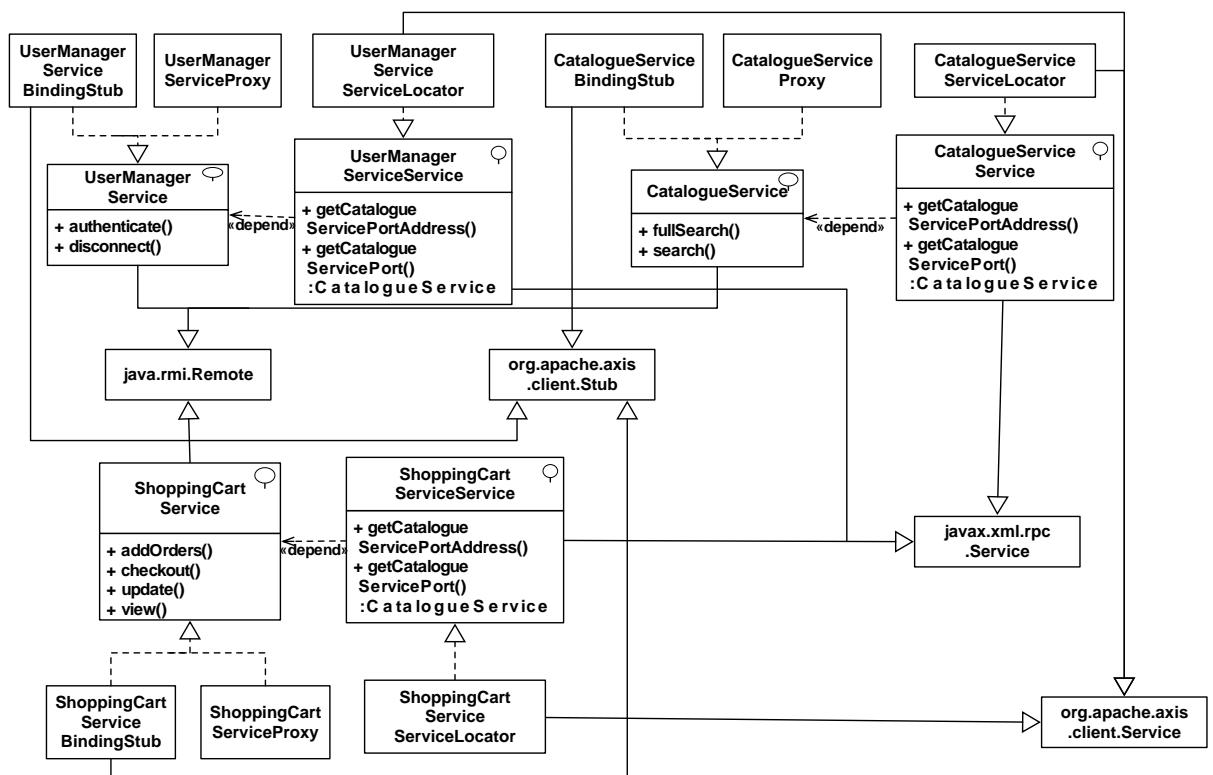


Figura 2.14. g8.bookshop.business.ws: diagramma completo dei web service

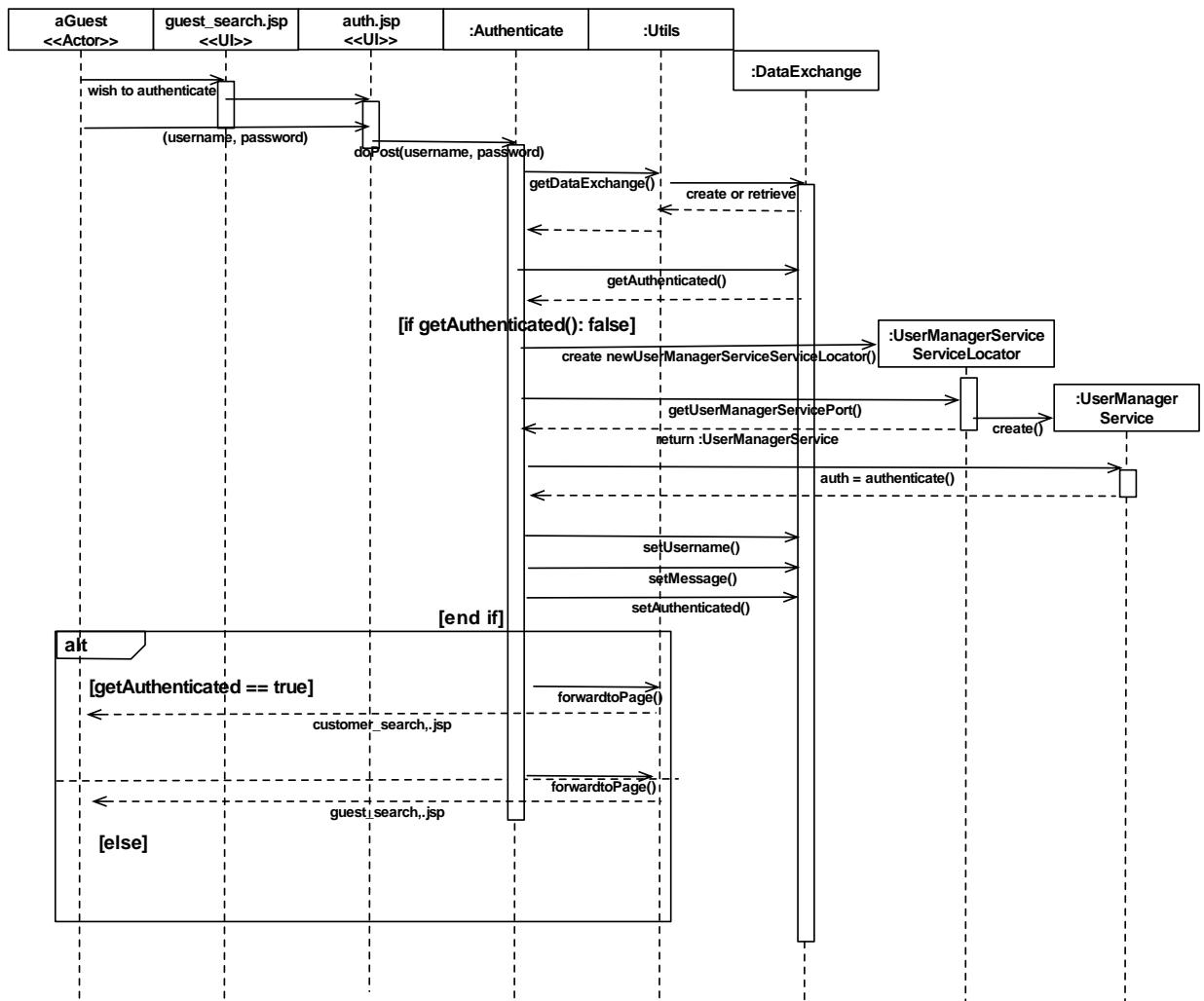


Flusso

Si riporta in questa sezione un sequence diagram non esaustivo per quanto riguarda la logica del lato presentation, ma utile per la comprensione di alcuni suoi funzionamenti fondamentali. Il caso d'uso

rappresentato è l'autenticazione, lo stesso che si è rappresentato relativamente al lato business: in questo modo speriamo di dare una visione complessiva sul sistema. In questo caso d'uso non viene effettuata la trasformazione XSLT operata internamente al POJO DataExchange, ma non ci è parsa un'operazione abbastanza significativa da necessitare ad ogni costo di essere rappresentata.

Figura 2.15. Diagramma di sequenza per il caso d'uso Autenticazione



Il caso d'uso viene innescato quando l'utente, trovandosi sulla pagina `guest_search.jsp`, notifica la sua volontà di autenticarsi selezionando un collegamento apposito. Viene di conseguenza caricata la pagina `auth.jsp`, nel quale si trova la maschera di autenticazione. L'utente vi inserisce il nome utente e la password e selezionando "submit" questi valori vengono passati alla servlet incaricata dell'autenticazione, `Authenticate`.

La servlet in primo luogo verifica che l'utente non già autenticato per il sistema: questa informazione può trovarsi solo all'interno dell'oggetto condiviso tra servlet e JSP, `DataExchange`. La servlet quindi richiede, attraverso un metodo della classe `Utils`, l'oggetto `DataExchange`: le viene restituito un nuovo oggetto qualora nessuno esistesse, o l'oggetto già esistente. A questo oggetto essa richiede lo stato di autenticazione dell'utente.

Qualora l'utente non risultasse autenticato la servlet provvede a contattare il web service: per farlo, crea un nuovo oggetto `ServiceLocator` al quale poi richiede la creazione di un oggetto `UserManagerService`. A questo oggetto fa la sua richiesta di autenticazione. La richiesta raggiunge il lato business dove viene completata: questa parte non è relativa al lato presentation e pertanto non è rappresentata nel diagramma. Quel che è rappresentato è il risultato della computazione

che ritorna alla servlet Authenticate, la quale conseguentemente setta opportuni parametri indicanti il nuovo stato di autenticazione nell'oggetto condiviso, nonché un messaggio testuale indicante il risultato dell'autenticazione (completata, o fallita).

Si verifica nuovamente lo stato dell'autenticazione, e attraverso il metodo `forwardToPage` della classe `Utils`, si trasferisce il controllo alla JSP appropriata: se l'utente risulta non autenticato, a `guest_search.jsp`, oppure a `customer_search.jsp`. La JSP chiamata leggerà le informazioni dall'oggetto condiviso, in particolare il messaggio sul risultato dell'autenticazione, che visualizzerà adeguatamente nella pagina.

2.4. Livello Business

2.4.1. Introduzione alle scelte progettuali

Come già chiarito nelle precedenti sezioni, la progettazione dell'intera applicazione è stata fortemente guidata dalla strutturazione a 4 livelli che le stesse specifiche Java EE suggeriscono. E' stato quindi naturale identificare l'area di ingerenza del livello business nel dominio di lavoro dell'applicazione di negozio virtuale di libri, delegando così ad altri livelli il compito di presentare i contenuti. Tuttavia quella che sembrerebbe una decisione scontata non è in realtà così diffusa: per ragioni d'opportunità si assiste, nella pratica comune, ad un accoppiamento eccessivo tra i due livelli. Infatti, in applicazioni simili, la gestione del carrello degli acquisti è spesso ad appannaggio esclusivo del livello web quando invece sarebbe chiara competenza del livello business. Questa accurata distinzione delle responsabilità ha permesso di ottenere due risultati progettuali molto significativi:

1. alta coesione all'interno degli stessi livelli;
2. basso accoppiamento tra i livelli.

La scelta di definire un formato XML per i messaggi di scambio tra la presentazione e il business unitamente all'utilizzo dei web service, ha reso ancor più indipendenti le due componenti.

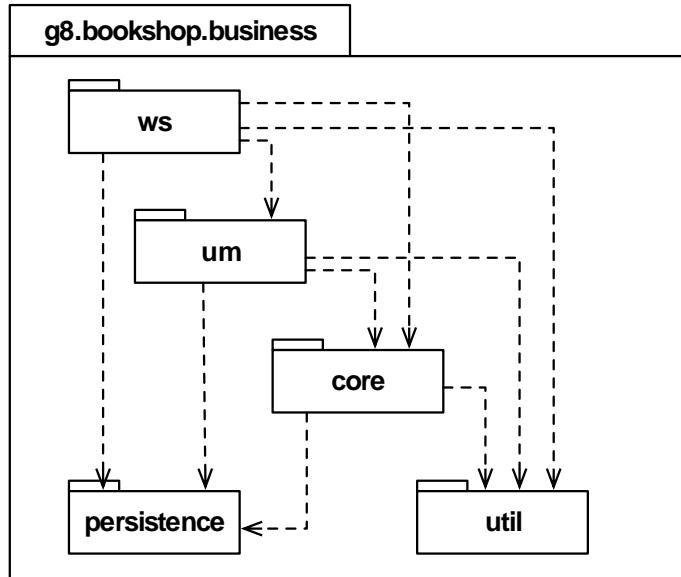
Un'attenzione particolare durante la progettazione è stata posta anche per quanto riguarda gli aspetti connessi alle prestazioni e alle tolleranze ai guasti. JBOSS AS 5.1.GA offre in questo senso importantissimi ausili tecnologici che hanno permesso di ottenere componenti ad alta disponibilità, quanto più replicati fra i nodi del cluster.

2.4.2. Scelte architetturali

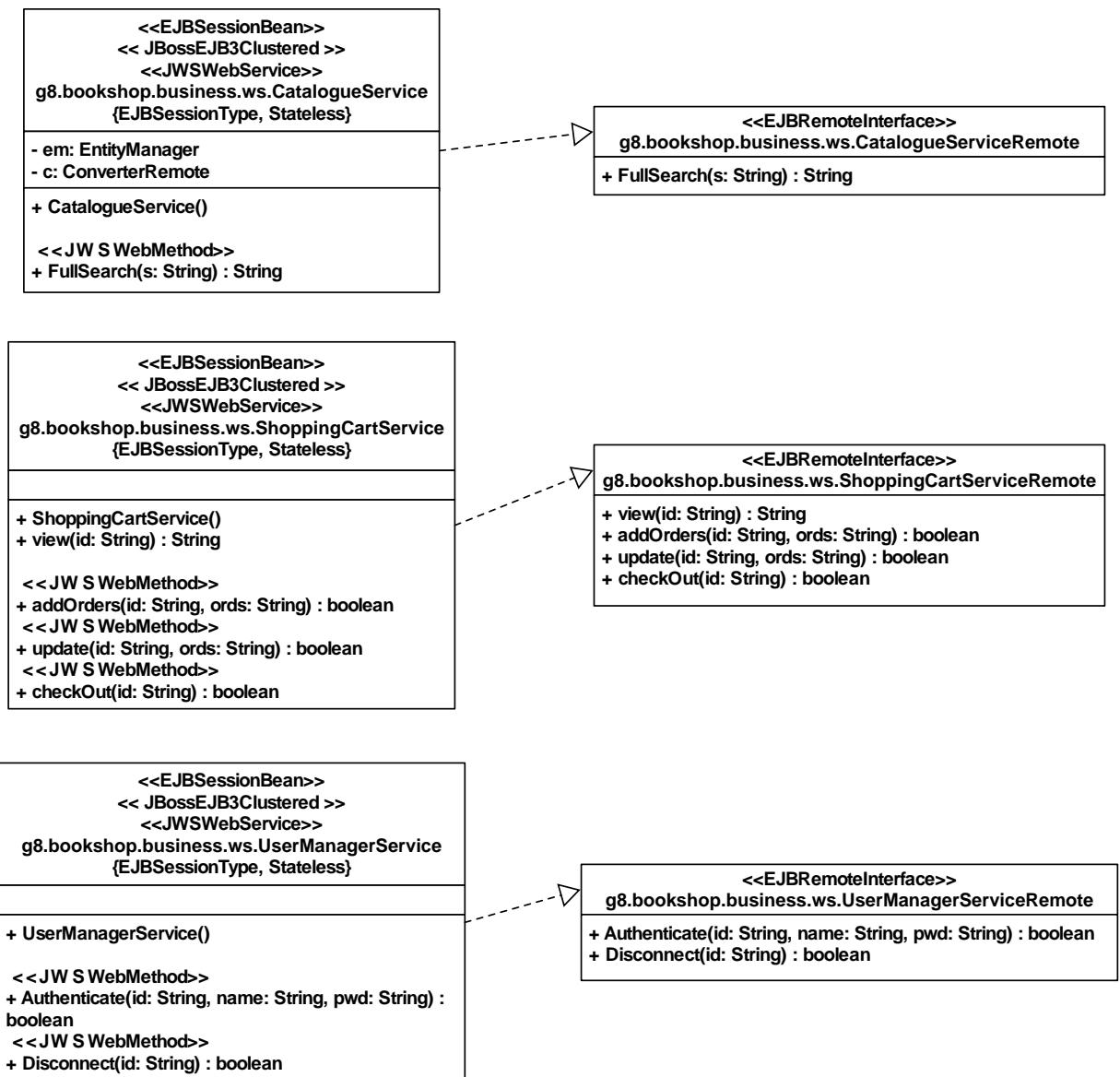
Il livello business è stato progettato secondo i criteri di modularità che fino ad ora hanno contraddistinto l'applicazione. Prima di scendere nei dettagli architetturali è opportuno chiarire alcuni aspetti chiave al fine di apprezzare al meglio la soluzione proposta. Allo stato attuale dell'arte, le uniche EJB di sessione esponibili tramite web service sono le EJB stateless. Questo fatto ha comportato alcune complicazioni alla progettazione del livello business dal momento che è richiesto il mantenimento di alcune informazioni di stato, come ad esempio il carrello. Nella fattispecie si è resa necessaria la realizzazione di un servizio singleton, il cui stato fosse sia condiviso tra i diversi client, sia preservato tra le diverse richieste. Nel prosieguo il lettore troverà spiegazioni più dettagliate.

In Figura 2.16, « g8.bookshop.business » si può osservare la strutturazione in pacchetti del livello e le relazioni di dipendenza che intercorrono fra essi. L'elenco sottostante chiarisce le responsabilità per ciascun modulo:

- g8.bookshop.business.ws: espone tramite web service stateless i servizi offerti dal livello business;
- g8.bookshop.business.um: gestisce gli utenti e ne mantiene lo stato nel negozio virtuale;
- g8.bookshop.business.core: definisce le entità, in senso lato, del business congiuntamente alle operazioni che le coinvolgono;
- g8.bookshop.business.persistence: costituisce il punto di accesso verso il livello informativo (contiene tutte le entity bean);
- g8.bookshop.business.util: offre alcune funzioni di utilità per tutto il livello business.

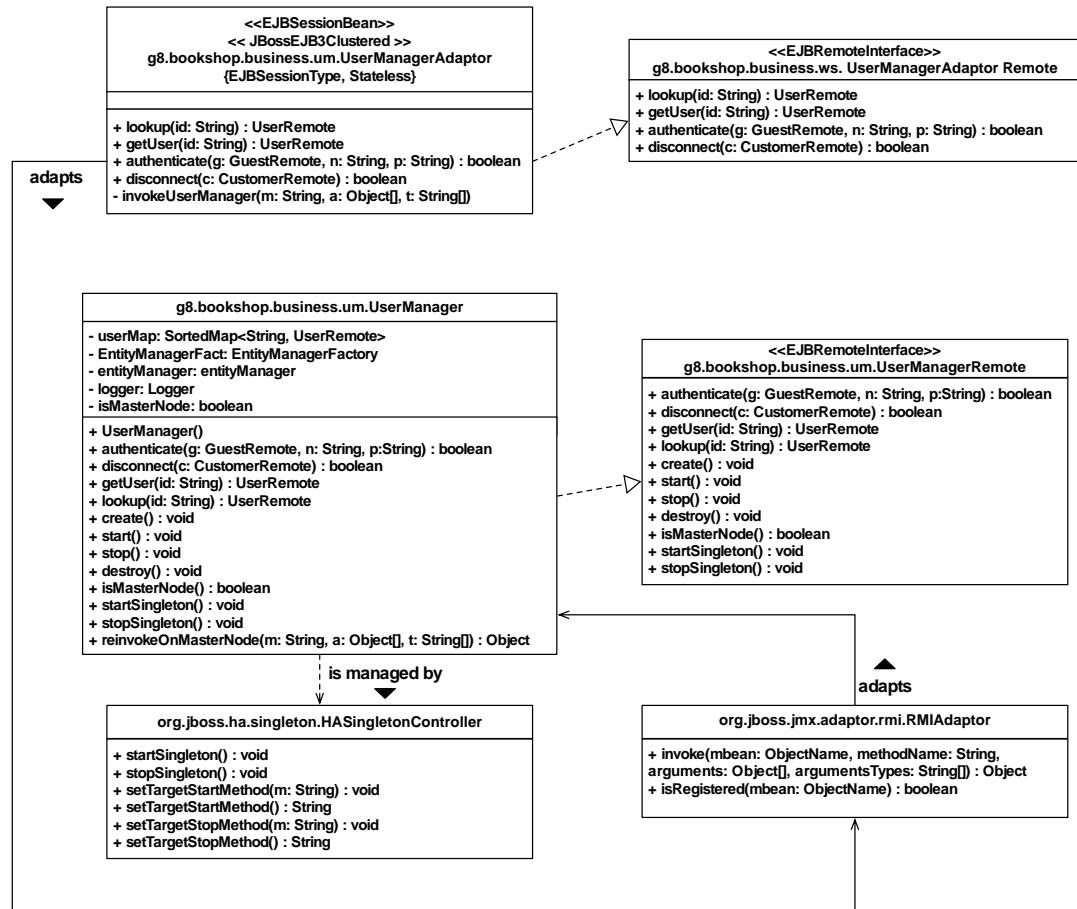
Figura 2.16. g8.bookshop.business

Scendendo un po' più in profondità, si osserva che il pacchetto `g8.bookshop.business.ws` è composto da tre EJB stateless, tutte esposte come web service: `CatalogueService`, `ShoppingCartService` e `UserManagerService`. La prima fornisce i metodi per l'accesso al catalogo dei libri (e.g. ricerca), la seconda al carrello dei clienti e infine la terza offre i servizi di autenticazione e disconnessione.

Figura 2.17. g8.bookshop.business.ws

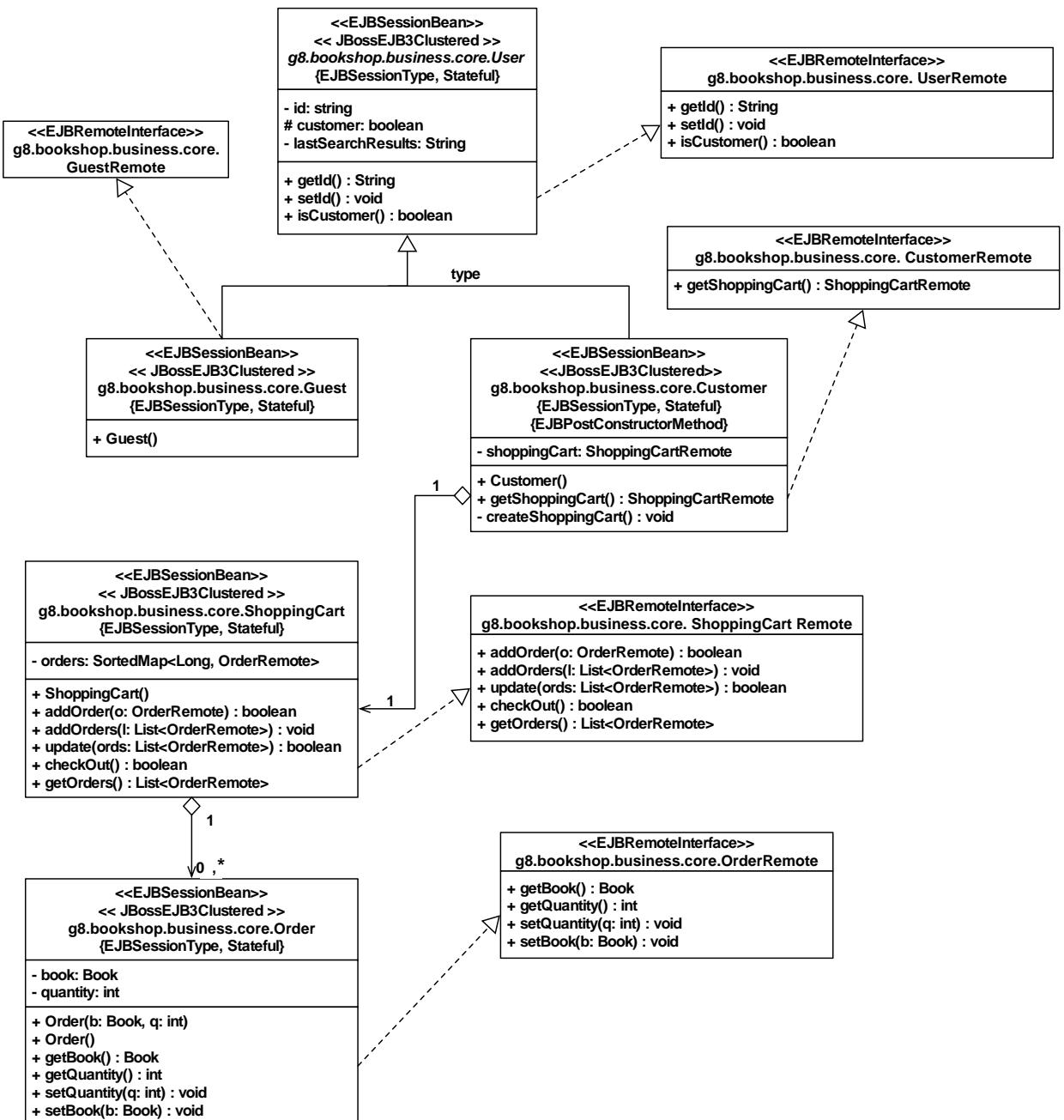
Il pacchetto `g8.bookshop.business.um` contiene la classe `UserManager` che implementa il servizio singleton per la gestione degli utenti e del loro stato. In particolare tramite una `HashMap` mantiene gli utenti che sono al momento presenti nel negozio virtuale. L'`HashMap` in questione associa a ciascun identificativo di sessione un utente. Come si vedrà nel seguito, il carrello è mantenuto all'interno della EJB che rappresenta il cliente. Infine il gestore degli utenti è anche il responsabile dell'autenticazione e della disconnessione dei clienti.

L'EJB `UserManagerAdaptor`, invece, fornisce metodi convenienti per accedere ai servizi offerti dal gestore degli utenti.

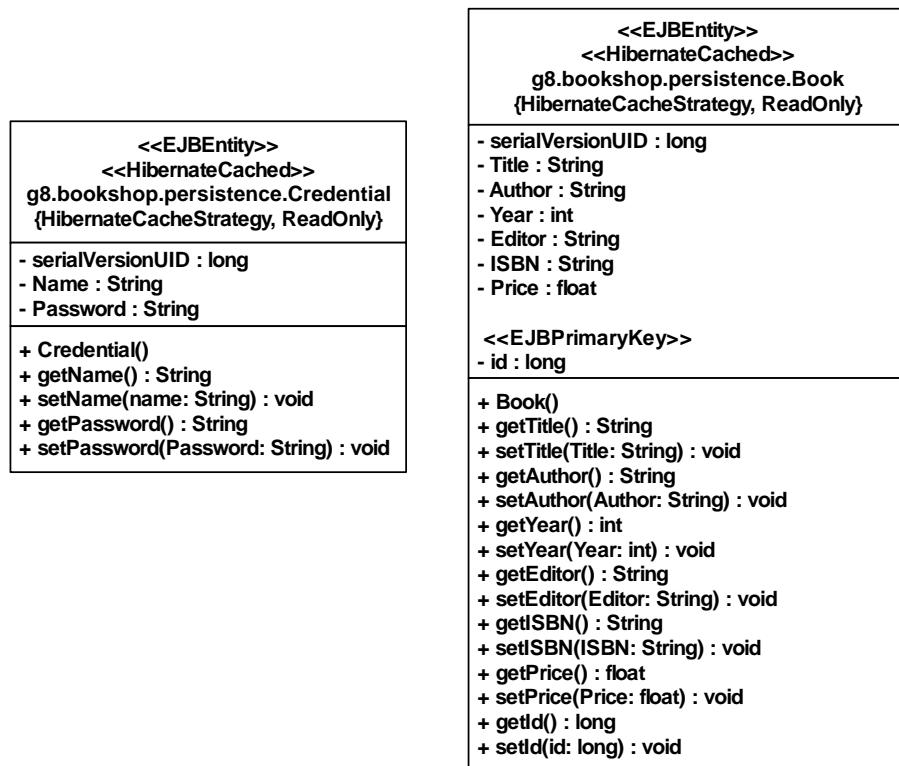
Figura 2.18. g8.bookshop.business.um

Il pacchetto `g8.bookshop.business.core` contiene tutte le EJB che rappresentano le entità del business che non hanno necessità di essere persistenti ma che comunque devono mantenere uno stato. In particolare si è codificato tramite `Guest` e `Customer` le due diverse tipologie di clienti che possono accedere al negozio virtuale di libri, cioè il visitatore occasionale e il cliente (per il quale è prevista l'autenticazione). Chiaramente solo il cliente possiede il carrello (`ShoppingCart`) che è costituito da un insieme di ordini (`Order`), cioè una coppia libro, quantità.

Figura 2.19. g8.bookshop.business.core

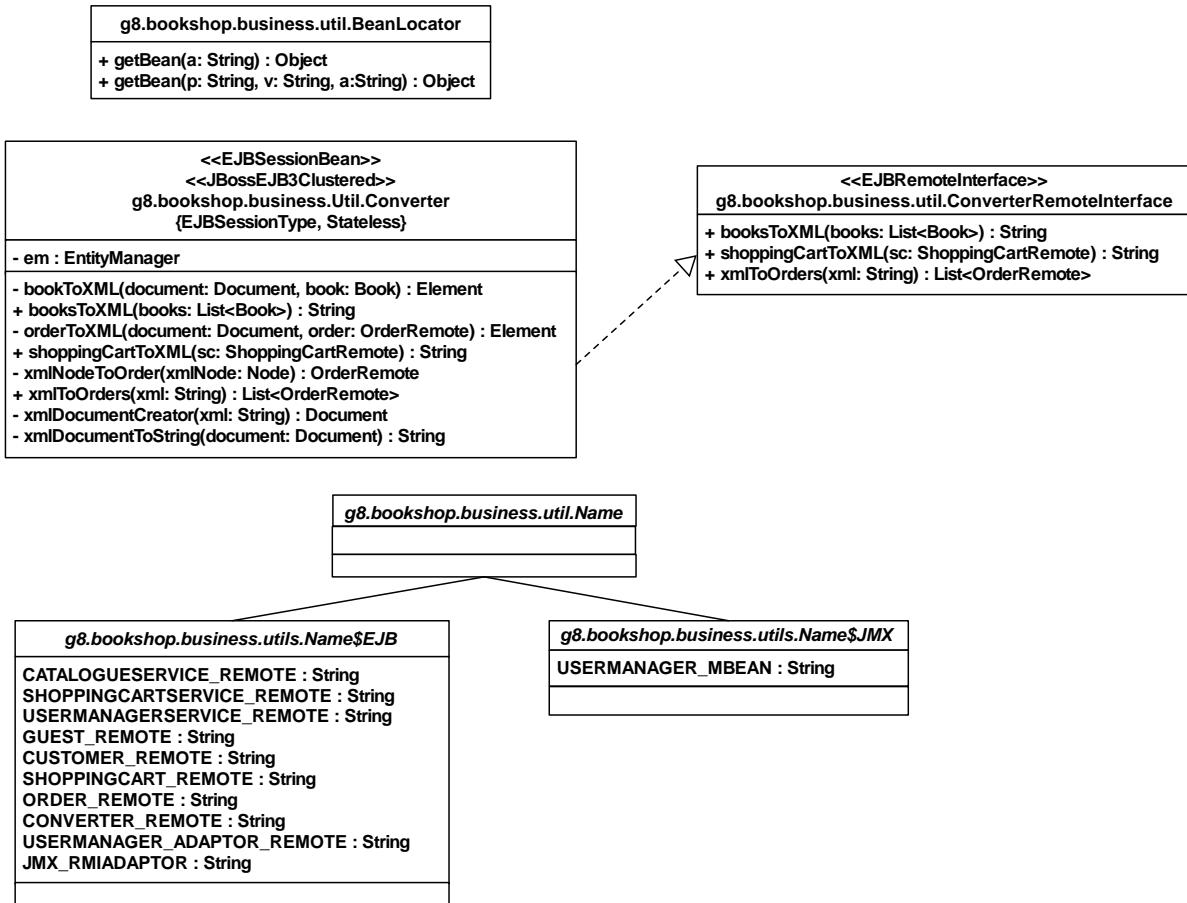


Il package `g8.bookshop.business.persistence` contiene le entità business che hanno necessità di essere persistenti. Nella fattispecie si tratta solamente di `Book` e `Credential`. La prima rappresenta l'entità libro mentre la seconda memorizza l'associazione tra nome utente e password, informazioni indispensabili per l'autenticazione dei clienti.

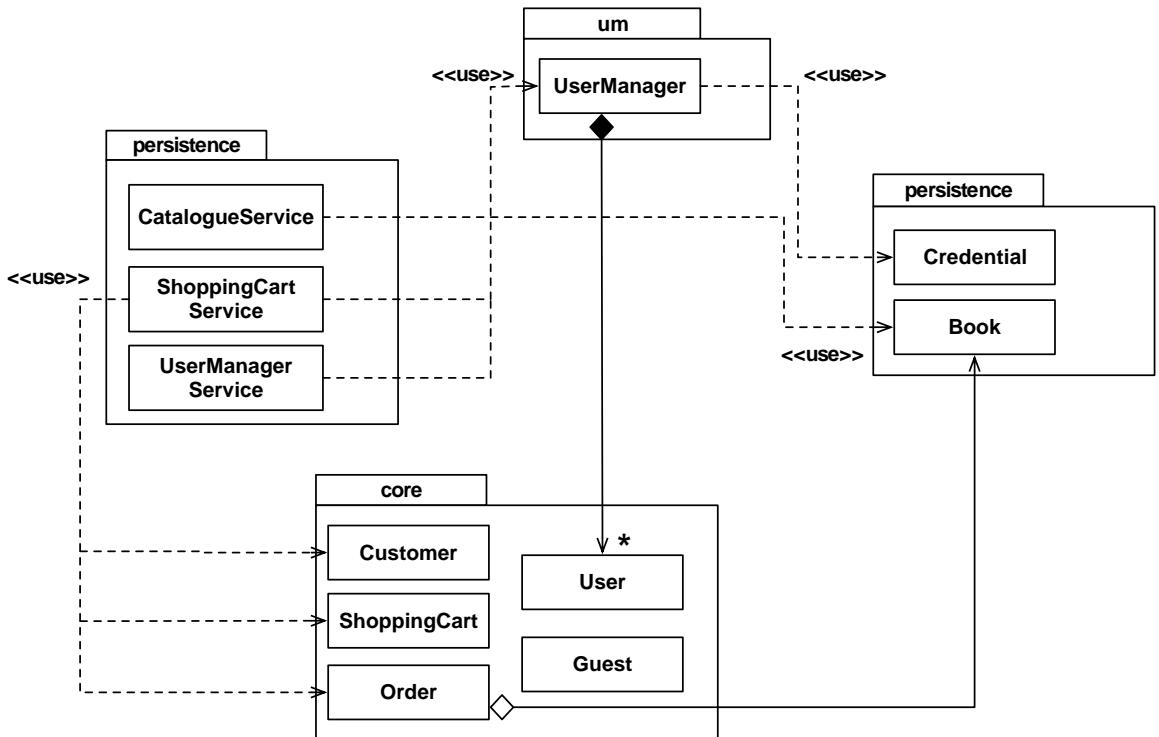
Figura 2.20. g8.bookshop.business.persistence

Infine il pacchetto `g8.bookshop.business.util` contiene alcuni strumenti di utilità per tutto il livello business. Il `Converter` è una EJB stateless che offre funzioni per la conversione da XML a oggetti business e viceversa; il `BeanLocator` offre una comoda primitiva per la localizzazione delle EJB e `Name` definisce alcune costanti utilizzate diffusamente nel progetto.

Figura 2.21. g8.bookshop.business.util



Il diagramma in Figura 2.22, « g8.bookshop.business » esplicita in maniera ancora più dettagliata le dipendenze tra i vari pacchetti, chiarendone la natura:

Figura 2.22. g8.bookshop.business

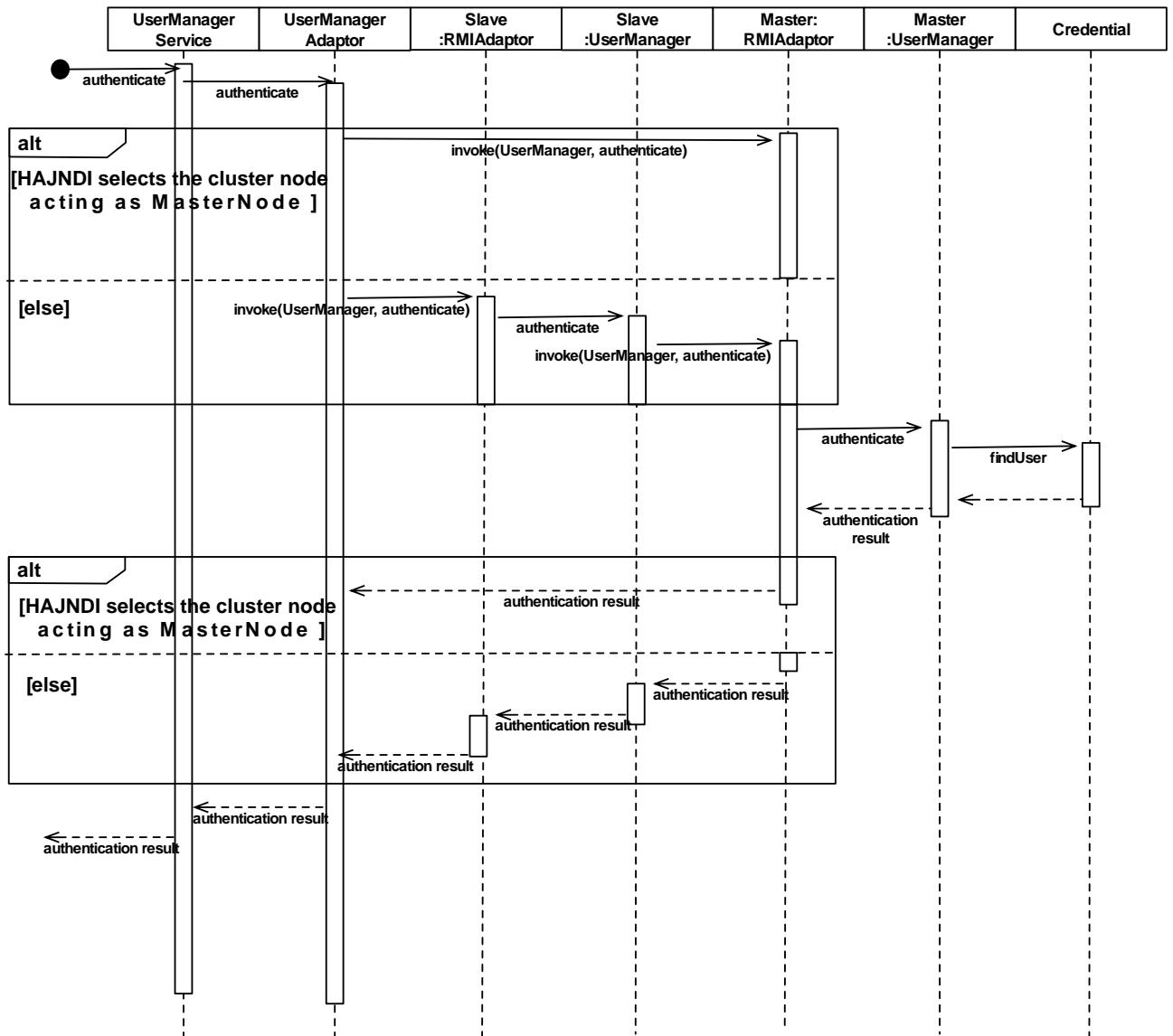
2.4.3. Scelte di schieramento

Servizio singleton ad alta disponibilità

La decisione di realizzare un servizio singleton per mantenere le informazioni di stato dei vari utenti rappresenta a prima vista un singolo punto di guasto per l'intera applicazione. Fortunatamente JBoss 5.1.0.GA offre l'opportunità di realizzare servizi singleton ad alta disponibilità. Nella fattispecie è possibile schierare il servizio su tutte le macchine del cluster ma eseguirlo solamente su uno, il master node. Nell'ipotesi in cui il master node si guastasse, il cluster ne rieleggerebbe uno nuovo sul quale sarebbe avviato nuovamente il servizio. Purtroppo al momento in cui si scrive, lo stato del servizio non viene preservato durante la procedura di migrazione sebbene s'intravedono gli spiragli per raggiungere anche questo traguardo.

Per schierare il servizio singleton ci si è appoggiati sull'**HASingletonController**, un servizio offerto da JBoss e attivo su ogni nodo del cluster. Il compito dell'**HASingletonController** è quello di monitorare la partizione e determinare se il nodo sul quale è attivo è il master node. A seconda della condizione del nodo, quindi, avvia o arresta il servizio invocando opportuni metodi (in questo caso dello **UserManager**). L'**HASingletonController**, comparato ad altre soluzioni analoghe, permette la più rapida gestione del failover in quanto il servizio è già schierato su tutte le macchine del cluster.

L'accesso al servizio singleton è però un po' complessa dal momento che non è una normale bean EJB 3.0 ma una mbean; il diagramma in Figura 2.23, « Diagramma di sequenza » lo dettaglia in maniera approfondita nel caso dell'autenticazione:

Figura 2.23. Diagramma di sequenza

Alta replicazione

Come annunciato nell'introduzione alle scelte progettuali, si è cercato di replicare quanto più possibile le componenti del business per avere benefici sia in termini di prestazioni che di tolleranza ai guasti.

In particolare tutte le entity bean sono state replicate grazie all'utilizzo della cache di secondo livello offerta da hibernate. Considerata che la natura degli accessi alle informazioni persistenti è di sola lettura, è stata impostata un politica di caching che favorisse le prestazioni in questo senso.

Grazie a HAJNDI, invece, si è ottenuto bilanciamento di carico per tutte le session EJB. Le stateful EJB inoltre sono state distribuite utilizzando la funzionalità "clustered" offerta da JBoss.

2.4.4. Scelte progettuali minori

La manutenibilità del prodotto è stata ricercata anche attraverso piccole scelte progettuali all'interno del livello business.

E' stato utilizzato il pattern Adapter per encapsulare le richieste allo `UserManager` in modo tale che dall'esterno fosse accedibile in maniera conveniente tramite una banale lookup.

Attraverso il `BeanLocator` invece si è fattorizzata la funzionalità di ricerca delle EJB evitando di duplicare eccessivamente il codice e, quindi, semplificandone la gestione in un unico punto.

2.5. Livello informativo

Il documento dei requisiti non impone particolari vincoli sul livello informativo che quindi non ha rappresentato un grosso scoglio nella realizzazione del progetto. Tuttavia è bene documentare alcune scelte che sono state operate in questo ambito.

In primis, si è scelto di non adottare il DBMS predefinito (Hypersonic) distribuito con JBoss AS 5.1.0.GA in favore del ben più noto e diffuso MySQL (versione 5.0.1), disponibile sulle macchine dipartimentali. La ragione di tale scelta è da ricercare nel fatto che Hypersonic non implementa correttamente le proprietà acide, esponendo quindi il database a possibili stati non consistenti. Inoltre le decisione di adottare MySQL ci ha permesso di familiarizzare ulteriormente con l'ambiente di JBoss, uno degli obiettivi principali del progetto.

In secundis, si è scelto di mantenere la progettazione del database il più semplice possibile lasciando aperta la possibilità di investire il tempo a disposizione in altre aree giudicate più interessanti, come lo schieramento, la progettazione architetturale e i servizi ad alta disponibilità.

Capitolo 3. Piano di processo

Prima di poter presentare il piano di processo che ha guidato lo svolgimento del progetto, è bene soffermarsi anzitutto ad illustrare sinteticamente gli obiettivi che ne hanno ispirato le scelte, poi a chiarire le modalità con le quali la pianificazione è stata sviluppata e gestita. Quando, quindi, saranno illustrati il piano preventivo e quello consuntivo, si potrà proseguire ad esporre alcune considerazioni di verifica della pianificazione.

3.1. Obiettivi della gestione di processo

Sia nella gestione di progetti software tramite metodologie ingegneristiche, sia nella realizzazione di progetti middleware con la tecnologia JBossAS i membri della squadra non potevano contare su quasi alcuna esperienza pregressa. Questo fatto si è previsto potesse costituire rischio ad alto fattore a cui essere sottoposti durante l'intero processo. Per questa ragione si è quindi scelto di adottare un modello di processo che facilitasse la gestione di cambiamenti durante il processo, che aderisse quindi ai principi dello *sviluppo agile* secondo le linee guida della Agile Alliance.

Le specifiche di processo hanno tuttavia imposto la produzione di una serie di *artefatti di processo* che, a causa dell'alto grado di dettaglio, risultano essere adatti a modelli di processo più dettagliatamente e rigidamente pianificati. È pertanto risultato preferibile un modello di processo che goda della produzione di tali artefatti.

Il buon livello comunicativo della squadra di lavoro, l'esiguità del numero dei suoi partecipanti, uniti agli obiettivi didattici del progetto hanno inoltre privilegiato un piano di processo che permettesse un *alto grado di partecipazione* della squadra alla gestione del processo. Il processo quindi avrebbe dovuto essere gestito tramite *artefatti semplici e concisi*, affinché possano essere compresi e utilizzati da ogni membro del team.

Queste tre caratteristiche hanno indirizzato la scelta verso un modello di processo recente, poiché è risalente al vicino 2005, ma piuttosto promettente. Si tratta di Agile Unified Process v1.1, pubblicato dalla Ambyssoft Inc. di Scott W. Ambler (<http://www.ambyssoft.com/unifiedprocess/agileUP.html>).

3.2. Gestione della pianificazione

Durante la prima fase di avvio del progetto (inception phase), si è abbozzato un piano di processo ad alto livello. Si è scelto di rendere il modello AUP ancora più agile, evitando le iterazioni che possono essere eseguite nelle fasi di elaborazione o costruzione (rispettivamente elaboration e construction) limitando il numero di iterazioni totali alle seguenti quattro:

1. Avvio
2. Elaborazione
3. Costruzione
4. Transizione

La stima dello sforzo esposta nel capitolo 7 ha consentito di stimare una data di conclusione del processo. Stimando quindi lo sforzo necessario ad ognuna delle fasi si è quindi ripartito il periodo temporale al fine di organizzare ad alto livello le attività del processo.

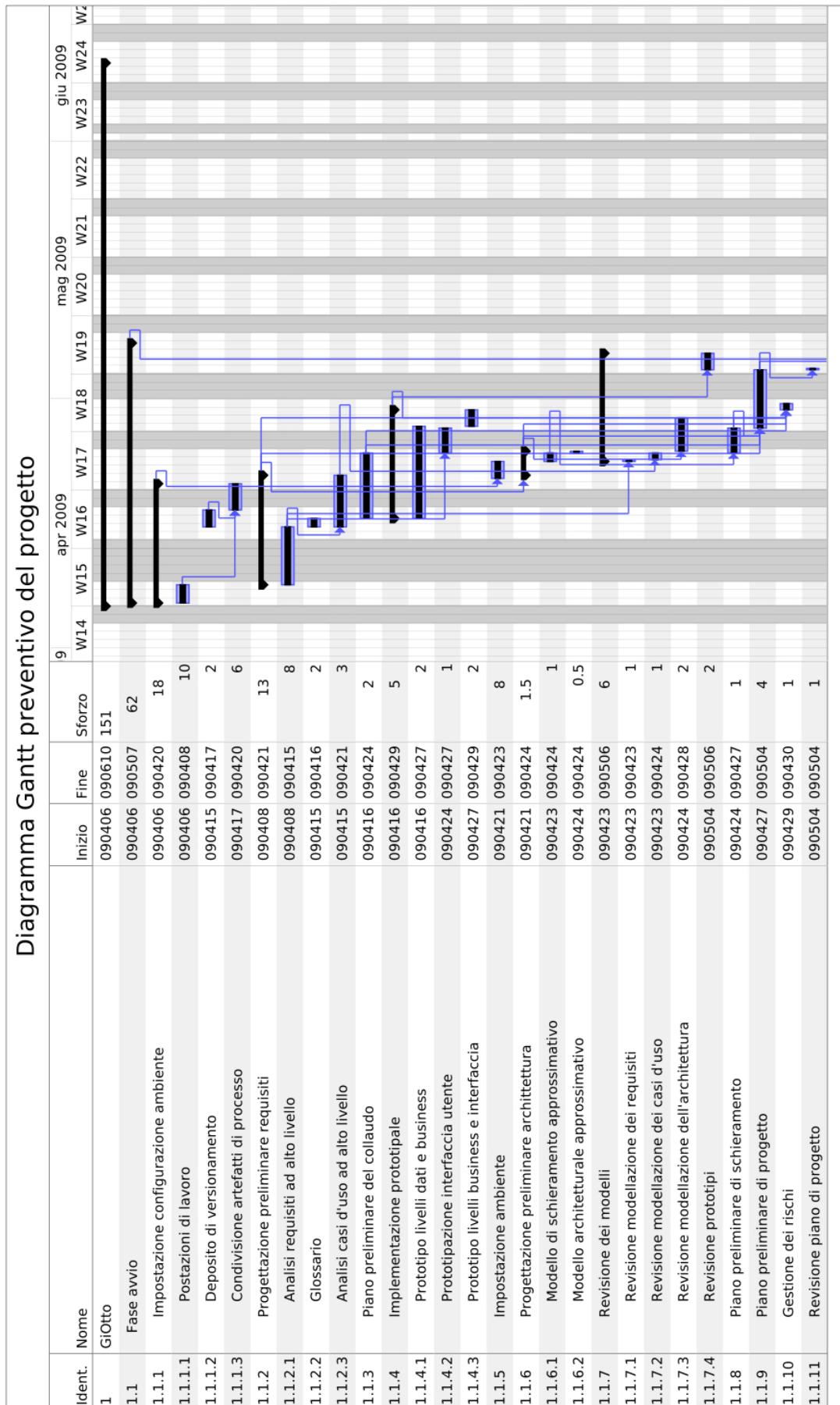
All'inizio di ogni fase si è proceduto alla specifica delle attività effettivamente da svolgere, in modo da poter organizzare il lavoro. Contestualmente si andavano anche ad allocare le risorse umane alle varie attività, secondo criteri di efficienza.

3.3. Diagrammi di pianificazione

La pianificazione è stata gestita e organizzata anche attraverso diagrammi di Gantt che potessero non solo illustrare visivamente la programmazione temporale delle attività, ma anche dettagliare con minuzia le relazioni di dipendenza che intercorrono fra di esse.

Di seguito si riporterà anzitutto il diagramma contenente il piano preventivo di processo, costruito unendo le pianificazioni realizzate all'inizio di ogni fase del processo. Poi si presenterà un rapporto dell'andamento effettivo del processo, realizzato nel diagramma di Gantt consuntivo del processo.

Diagramma Gantt preventivo del progetto

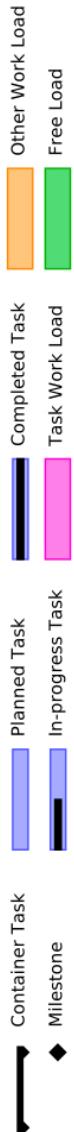
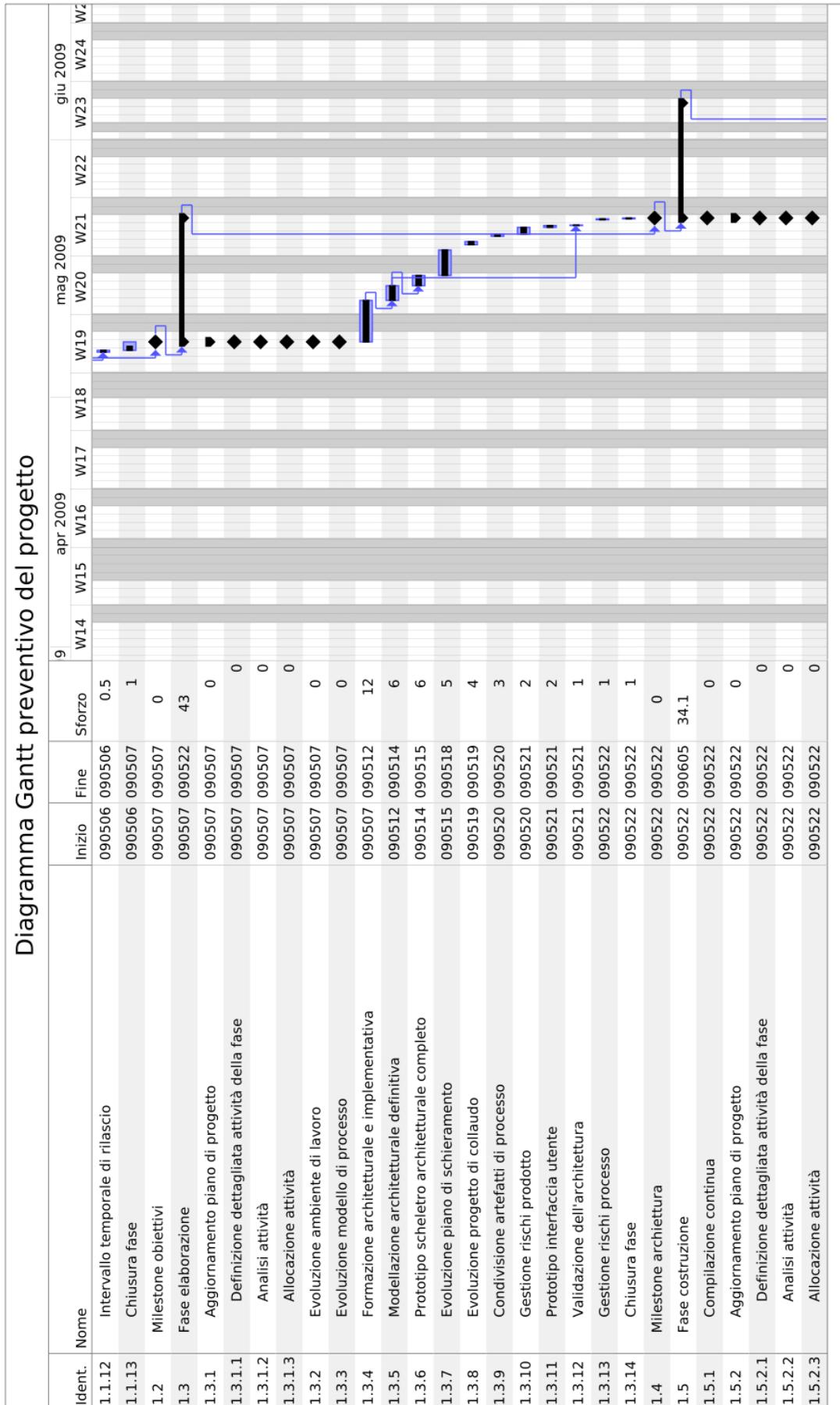


Page 1 of 4

090705

Generated on 09/07/05 with TaskJuggler 2.4.1

Diagramma Gantt preventivo del progetto

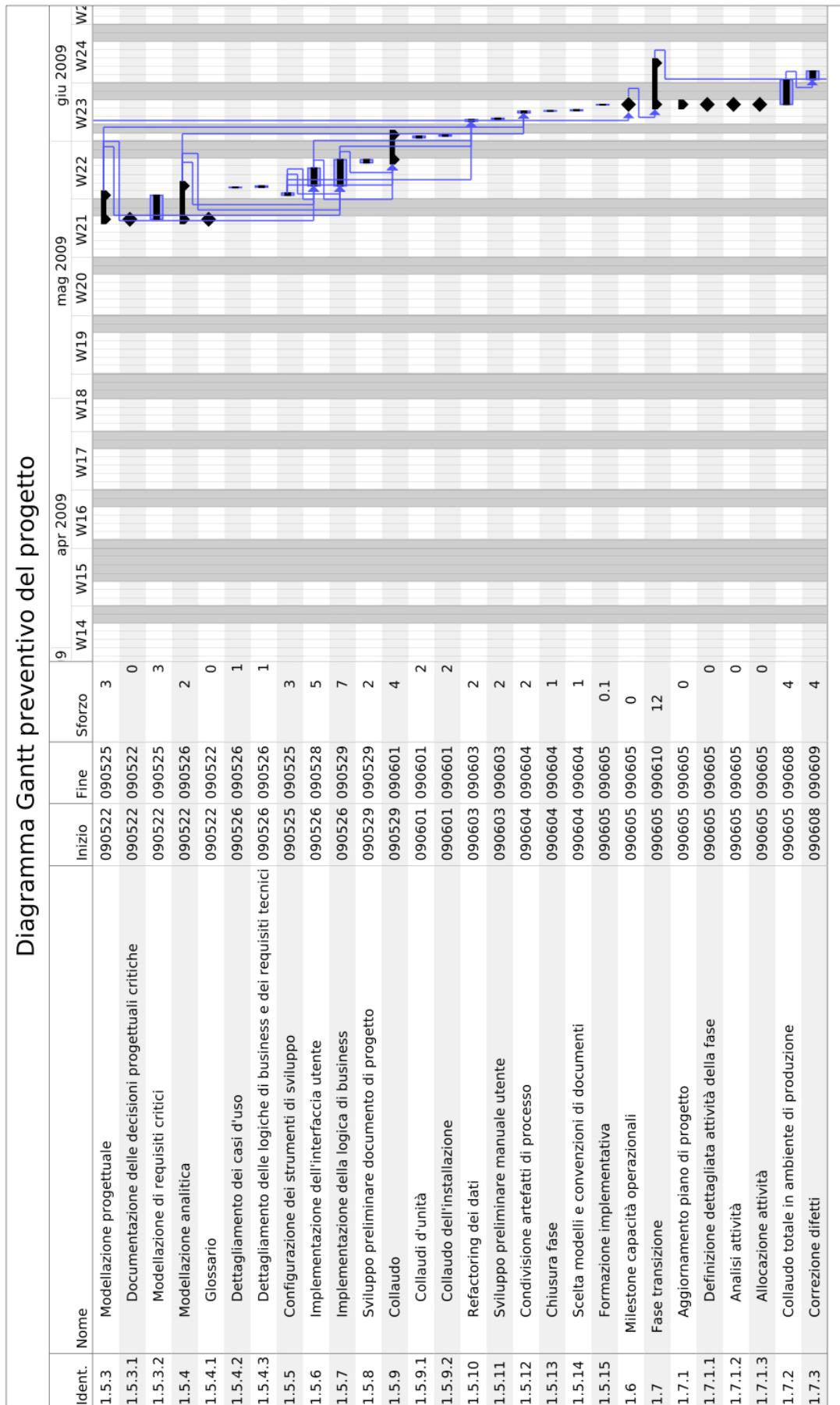


Page 2 of 4

090705

Generated on 09/07/05 with TaskJuggler 2.4.1

Diagramma Gantt preventivo del progetto



Page 3 of 4

090705

Generated on 09/07/05 with TaskJuggler 2.4.1

Diagramma Gantt preventivo del progetto

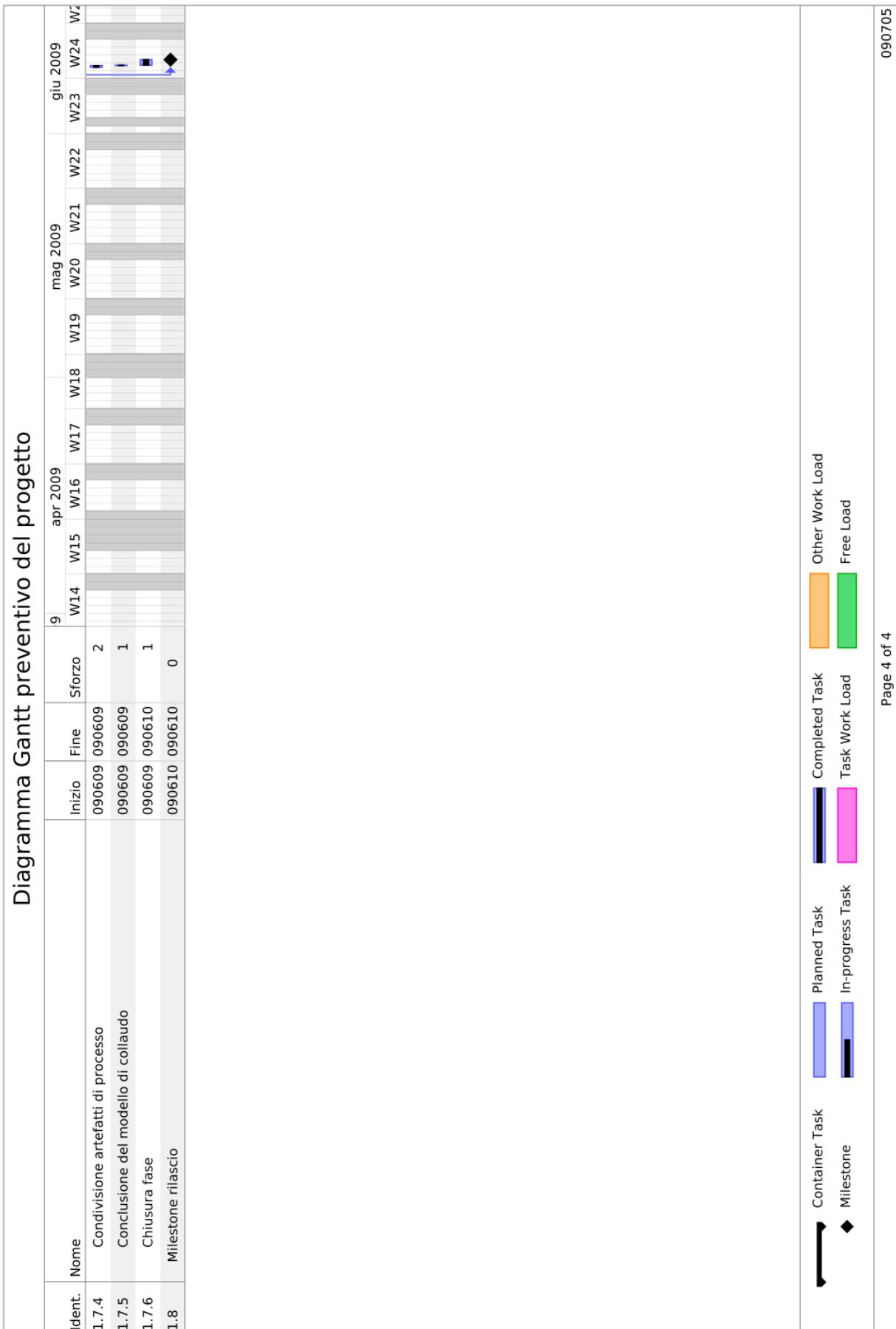
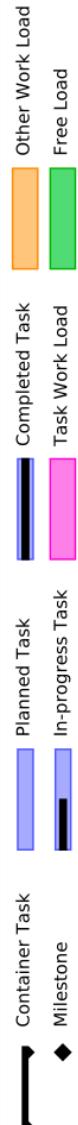
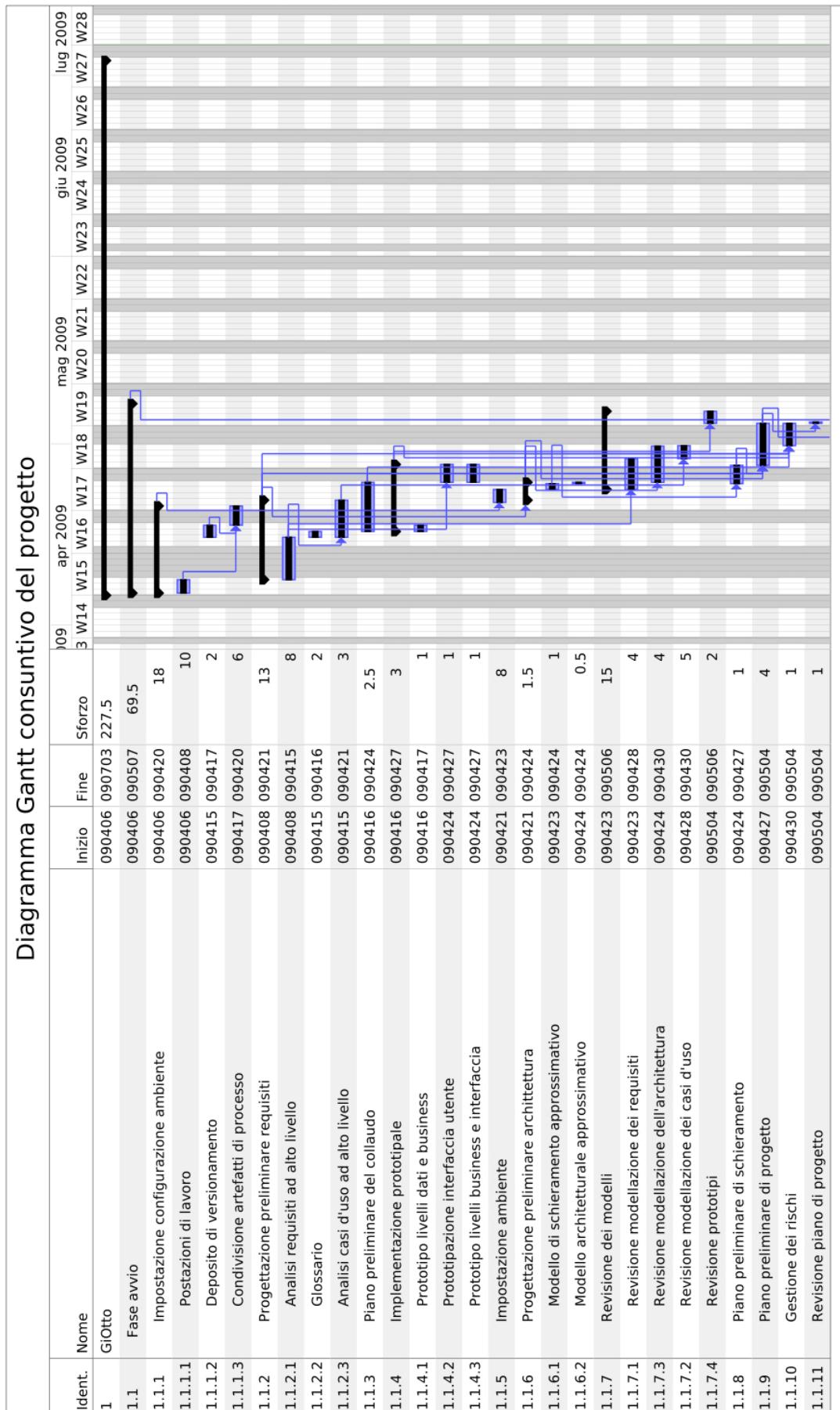
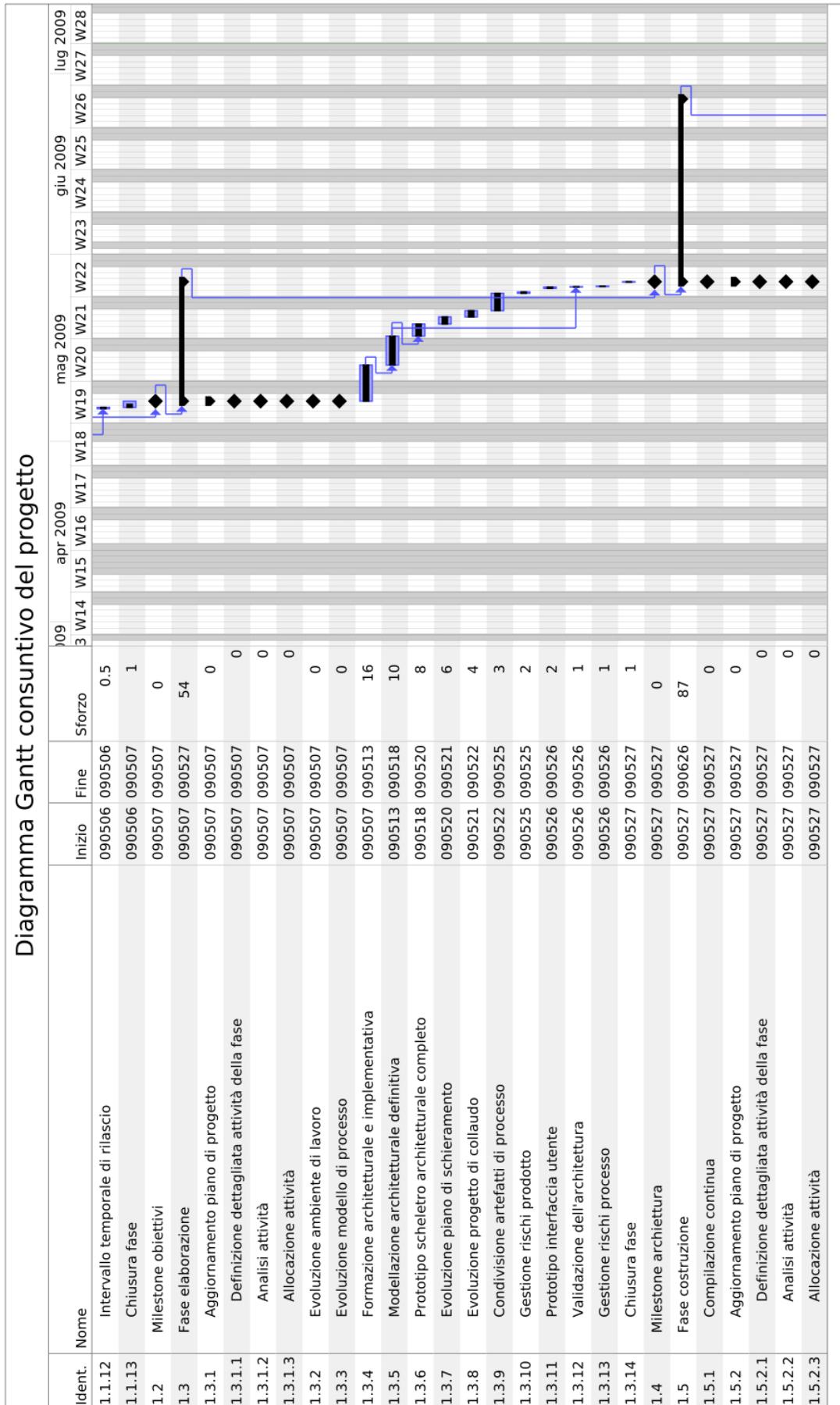


Diagramma Gantt consumutivo del progetto



Page 1 of 4

Diagramma Gantt consumutivo del progetto



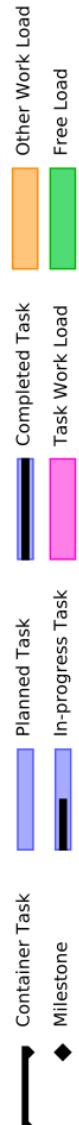
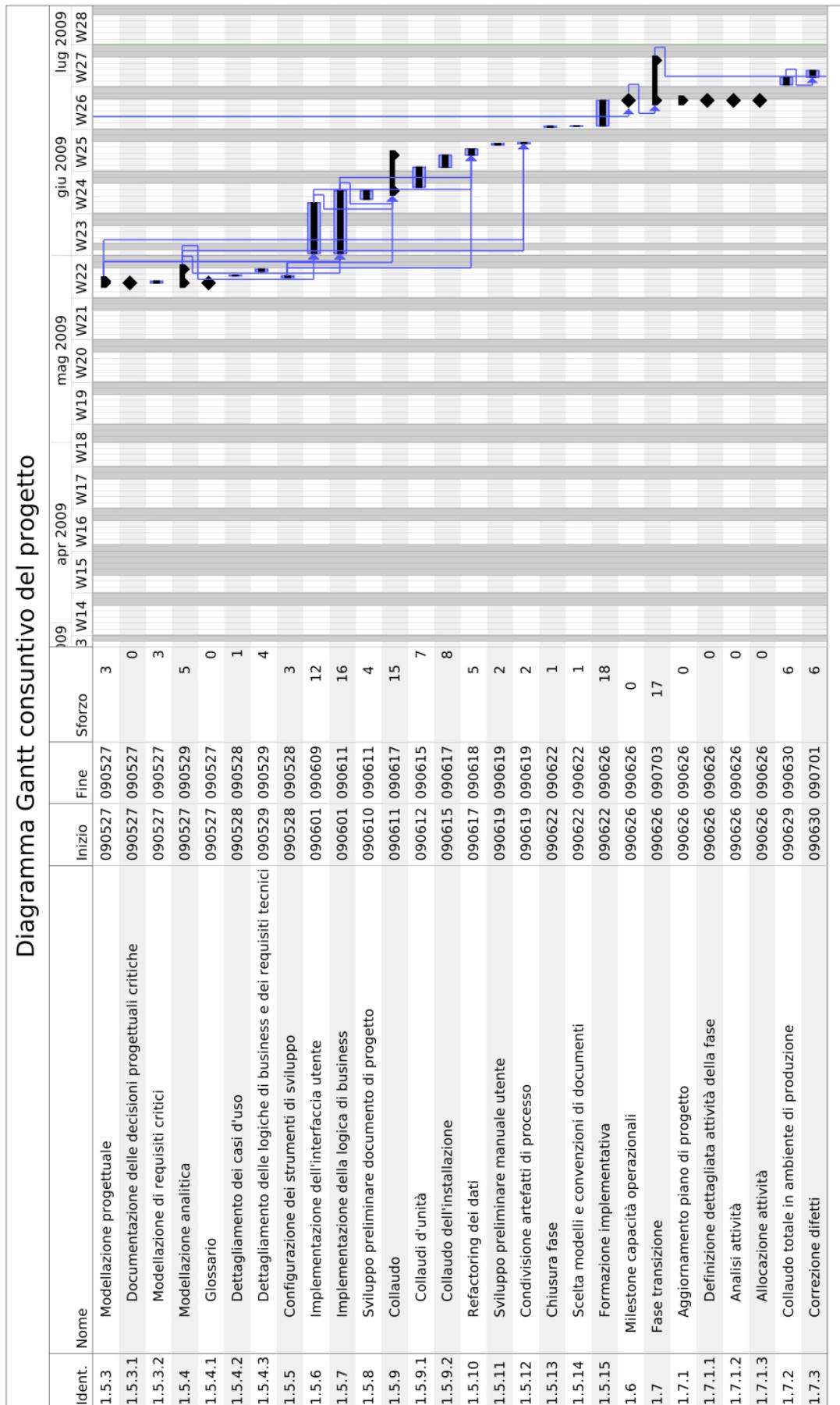
Container Task
 Planned Task
 In-progress Task
 Completed Task
 Milestone
 Milestone
 Other Work Load
 Task Work Load
 Free Load

Page 2 of 4

090705

Generated on 09/07/05 with TaskJuggler 2.4.1

Diagramma Gantt consumutivo del progetto

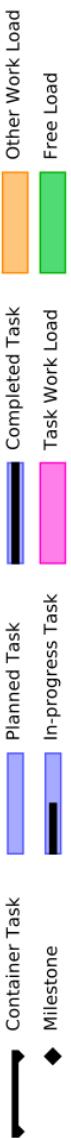
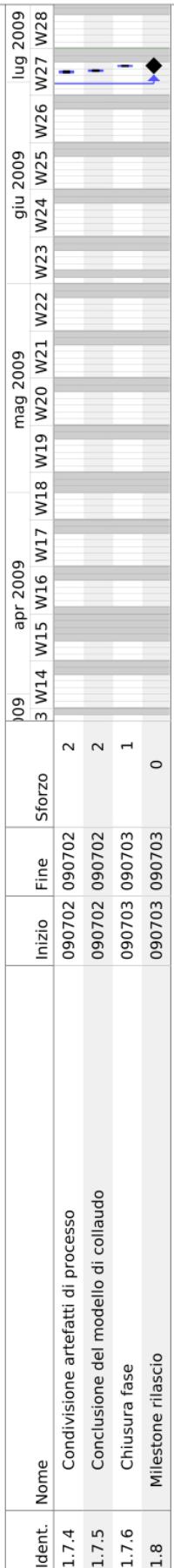


Page 3 of 4

Generated on 09/07/05 with TaskJuggler 2.4.1

090705

Diagramma Gantt consumutivo del progetto



Page 4 of 4

090705

Generated on 09/07/05 with TaskJuggler 2.4.1

3.4. Verifica della pianificazione

Giudicando la mera discrepanza di oltre due settimane fra il piano preventivo e quello consuntivo, la pianificazione non può che risultare grandemente perfettibile. Un'analisi più attenta rivela che gli errori di valutazione sono stati molto limitati e che quindi il piano è stato elaborato con una qualità più che discreta.

Il primo e più importante errore di valutazione i tempi da dedicare alla formazione implementativa, all'implementazione e al collaudo durante la fase di costruzione. Sottovalutando il rischio di non padroneggiare a sufficienza bene la tecnologia, si sono prese alcune scelte architettoniche (in particolare quelle trattate nella sezione 2.4.3) che hanno richiesto una formazione implementativa estremamente dettagliata dell'application server JBoss e non prevista per quella fase.

Il secondo errore riguarda più banalmente la mancanza di esperienza nella gestione di progetti secondo metodologie ingegneristiche che ha reso difficile l'effettuazione di stime e di pianificazioni. Si tratta di un rischio che, pur essendo chiaro, non poteva essere gestito completamente, ma solo affrontato di volta in volta, secondo gli scopi didattici del progetto.

Capitolo 4. Tool utilizzati

Questo capitolo elenca gli strumenti utilizzati durante lo sviluppo del progetto, sia per lo sviluppo stesso che per la documentazione. Non sono stati inseriti in questa trattazione gli strumenti lasciati al gusto de singoli, come ad esempio gli editor di testo per la documentazione semplice. Fanno eccezione particolari strumenti non utilizzati da tutti i componenti del gruppo ma tuttavia significativi nello sviluppo, le cui funzionalità non sarebbero state fornite allo stesso modo da un qualunque altro strumento simile: ad esempio, particolari editor di immagini o trasformatori XSLT.

4.1. Strumenti per la gestione del progetto

Google Groups

Riferimento	http://www.groups.google.com/
Licenza	L'utilizzo dei servizi forniti da Google Groups è gratuito. Le condizioni di utilizzo e la policy sulla privacy sono specifiche di Google e descritte in documenti visionabili online.
Descrizione	Servizio gratuito offerto da Google che permette di creare e gestire gruppi di discussione.
Motivazioni della scelta	Nessuna motivazione in particolare: una mailing list e alcuni servizi addizionali come la possibilità di caricare files in uno spazio comune e avere a disposizione un archivio delle mail scambiate era importante per la gestione dello sviluppo del progetto. Abbiamo scelto Google Groups per la familiarità coi servizi di Google in generale. Non ci risulta che tra gli altri servizi che forniscono funzionalità simile qualcuno si distingua per qualche aspetto particolarmente interessante per le nostre limitate necessità, di conseguenza nessuna ricerca in questo senso è stata fatta.

Google code

Riferimento	http://code.google.com/
Licenza	L'utilizzo dell'applicazione è possibile, e gratuito, per chiunque possiede un account Google. Le condizioni di utilizzo e la policy sulla privacy sono specifiche di Google e descritte in documenti visionabili online.
Descrizione	Per quanto riguarda i progetti caricati il sito permette di impostare una tra le seguenti nove licenze disponibili: Apache, Artistic, BSD, GPLv2, GPLv3, LGPL, MIT, MPL e EPL.
Motivazioni della scelta	Sito di Google per sviluppatori che intendano sviluppare progetti da rilasciare sotto una licenza free software. Fornisce diversi servizi, di nostro interesse principalmente il servizio gratuito di hosting con controllo delle revisioni mediante Subversion.

il codice anche tramite browser, e questo ci ha permesso di rendere i documenti della nostra relazione disponibili sul wiki al volo, senza la necessità di convertire in un altro documento da caricare ogni volta. In ultimo, anche la possibilità di effettuare issue tracking ci ha interessato.

Subclipse 1.6

Riferimento	http://subclipse.tigris.org/
Licenza	Eclipse Public Licence
Descrizione	Plug-in per Eclipse che fornisce supporto per Subversio internamente all'IDE.
Motivazioni della scelta	Subclipse è la soluzione standard per gestire un repository Subversion internamente a Eclipse. Abbiamo tutti concordato che avere SVN integrato nella nostra IDE di preferenza fosse molto comodo: questa scelta non è stata fatta inizialmente ma è risultata successivamente come dato di fatto. Subclipse permette di visualizzare i file modificati rispetto all'ultima sincronizzazione direttamente durante la fase di programmazione, e caricare le proprie modifiche è molto veloce in quanto è sufficiente cambiare prospettiva nell'editor. Anche grazie a questo è stato fatto un uso massiccio dell'SVN.

4.2. Strumenti per la documentazione

Docbook V5.0

Riferimento	http://www.docbook.org [http://www.docbook.org/]
Licenza	Oasis Open Standard
Descrizione	Linguaggio di markup semantico, basato su XML, per la redazione di documentazione tecnica. È possibile da sorgenti Docbook generare, tra gli altri, documenti html, xhtml, pdf, tex.
Motivazioni della scelta	Docbook è un linguaggio estremamente flessibile e versatile, ben più di LaTeX. Il vocabolario che offre è molto ricco, e grazie al fatto che è basato su XML la riorganizzazione della struttura complessiva di un documento redatto in DocBook si può effettuare rapidamente con un semplice foglio di stile XSLT. Permette minore precisione nella specifica delle caratteristiche tipografiche del documento finale rispetto a LaTeX, ma abbiamo ritenuto che la qualità dei documenti prodotti fosse comunque più che accettabile.
	Docbook ci ha inoltre permesso di mantenere una versione XHTML dei nostri documenti sempre aggiornata sul wiki, generata al volo alla visione delle pagine wiki a partire dai sorgenti Docbook sul SVN. Tramite dblatex è ad ogni modo possibile trasformare i sorgenti Docbook in LaTeX: modifiche specifiche su aspetti tipografici possono volendo essere effettuate qui.
	Nessuno dei membri del gruppo aveva utilizzato precedentemente Docbook, e proprio per questo abbiamo voluto utilizzarlo nello sviluppo di questo progetto,

considerando che sarebbe stata una buona occasione per imparare a utilizzarlo.

dblatax

Riferimento	http://dblatax.sourceforge.net/
Licenza	Oasis Open Standard
Descrizione	Si tratta di una collezione di fogli di stile che permettono di convertire sorgenti Docbook in diversi formati passando attraverso una prima conversione in TeX.
Motivazioni della scelta	Con dblatax abbiamo potuto trasformare la nostra documentazione scritta in Docbook in LaTeX. Mentre Docbook ci ha permesso grande maneggevolezza del contenuto, la conversione in LaTeX ci ha permesso controllo sulla versione finale per la stampa. Grazie a dblatax abbiamo potuto produrre file TeX, da modificare e compilare successivamente, o direttamente file PDF o PS.

4.3. Strumenti per lo sviluppo: application framework, IDE e editor

Eclipse JEE Ganymede

Riferimento	http://www.eclipse.org/
Licenza	Eclipse Public Licence
Descrizione	Eclipse è un ambiente di sviluppo software che comprende un IDE e un sistema di plugin che permettono di estendere l'IDE stessa. Si tratta di un ambiente multilinguaggio, ma indirizzato in primis agli sviluppatori Java. Nella sua versione EE si rivolge ovviamente allo sviluppo di applicazioni di tipo Enterprise, in particolare J2EE.
Motivazioni della scelta	Quando si tratta di sviluppare in codice Java, la scelta dell'IDE da utilizzare è molto spesso tra Eclipse e NetBeans. Abbiamo scelto il primo in favore del secondo in quanto era uno strumento già stato utilizzato da tutti i membri del gruppo, e una breve ricerca non ci ha indicato caratteristiche di NetBeans che potessero interessarci in modo particolare. Come diretta conseguenza di questa scelta, si sono utilizzati il plugin Subclipse per le interazioni col repository SVN, e il plugin JBossTools per una migliore integrazione dell'IDE con JBoss. Ogni componente del gruppo ha poi personalizzato l'IDE a suo piacimento con altri, o nessuno, dei numerosi plugin disponibili per essa.

JBoss Tools 3.0

Riferimento	http://www.jboss.org/tools
Licenza	GNU General Public Licence
Descrizione	JBoss Tools è un insieme di plugin per Eclipse per lo sviluppo di applicazioni JBoss e J2EE.

Motivazioni della scelta

JBoss Tools è una scelta ovvia per lo sviluppo con JBoss ed Eclipse. I plugin che lo costituiscono permettono di avviare, configurare e fermare in modo semplice e veloce JBoss. Offrono supporto alla modifica di file HTML e JSP, nonché di file di configurazione XML propri di JBoss o J2EE grazie a un editor visuale. Permettono la generazione automatica di test e servizi, e includono funzionalità per packaging efficiente e deployment di vari tipi di progetti.

4.4. Strumenti UML

Buona parte dei diagrammi UML è stata disegnata su carta o su lavagna, fotografata e riportata in formato elettronico solo successivamente: questo perché per quanto semplici e intuitivi fossero gli strumenti scelti per la progettazione UML, questi non eguagliavano l'immediatezza della bozza a mano libera durante gli incontri di gruppo. I diagrammi così schizzati sono poi stati riportati in relazione alle volte nella forma semplicemente fotografata, altre schizzati con OmniGraffle, altre ancora con GreenUML (queste ultime nei casi nei quali ci interessassero le funzionalità di ingegneria diretta e inversa di GreenUML).

Green UML 3.1

Riferimento

<http://green.sourceforge.net/>

Licenza

Eclipse Public Licence

Descrizione

Green è un editor UML che supporta sia software engineering che reverse engineering. Permette di creare diagrammi UML a partire da codice e di generare codice da una classe UML appena disegnata.

Motivazioni della scelta

Abbiamo ritenuto che Green UML fosse il miglior editor UML rilasciato con una public licence disponibile, e uno dei pochi a permettere ingegneria inversa. L'interfaccia è semplice, e le immagini prodotte sono esportabili in jpg e gif, quindi facilmente inseribili in documentazione e visualizzabili con qualsiasi visualizzatore di immagine. In ultimo, Green è sviluppato come plugin di Eclipse, caratteristica che abbiamo molto apprezzato in quanto ci ha permesso sia di non aumentare ulteriormente il numero di diverse applicazioni indipendenti da utilizzare allo stesso tempo, sia di sfruttare al massimo le sue funzionalità di produzione codice e ingegneria inversa.

OmniGraffle Professional 5

Riferimento

<http://www.omnigroup.com/applications/OmniGraffle/>

Licenza

Proprietaria

Descrizione

OmniGraffle è un'applicazione per disegno e diagrammi sviluppata per Mac Os X. È uno strumento molto intuitivo e molto veloce per la creazione di diagrammi anche complessi.

Motivazioni della scelta

OmniGraffle era stato utilizzato da uno dei componenti del gruppo in precedenza: nonostante si tratti di software proprietario e utilizzabile solo sulla piattaforma Mac Os X, è stato sfruttato molto per schizzare velocemente in virtù della sua immediatezza e facilità d'uso. Permette di esportare in molti formati le immagini prodotte, tra cui SVG e PDF, ottimali

per l'inserimento nella relazione Docbook. È inoltre facilmente estendibile con una serie di stencil di strutture UML.

4.5. Strumenti per la gestione del processo

TaskJuggler 2.4.1

Riferimento	http://www.taskjuggler.org/
Licenza	GNU Public Licence version 2
Descrizione	TaskJuggler è un software per il supporto alla gestione di progetto. Fornisce funzionalità di pianificazione, tracciamento degli obiettivi, assegnazione delle risorse, analisi dei rischi.
Motivazioni della scelta	Si tratta di un gestore di progetto molto evoluto, ricco di funzionalità, col quale abbiamo ottimamente gestito tutto quanto riguarda il piano di processo e l'analisi di costi e sforzi. È stato ritenuto il miglior software open source di questo tipo. Fornisce template già pronti per un inizio veloce, permette di generare e visualizzare diagrammi di Gantt con una comoda interfaccia grafica. Permette di esportare tutti i manufatti prodotti in formati facili allo scambio e alla conversione.

Google Calendar

Riferimento	http://www.calendar.google.com/
Licenza	L'utilizzo dell'applicazione è possibile, e gratuito, per chiunque possieda un account Google. Le condizioni di utilizzo e la policy sulla privacy sono specifiche di Google e descritte in documenti visionabili online.
Descrizione	Applicazione Web di tipo calendario di Google.
Motivazioni della scelta	Dal momento che tutti i membri del gruppo possedevano già un account Google, necessario per l'utilizzo del servizio, Google Calendar ci si è subito presentata come una semplice soluzione per la gestione degli incontri del gruppo, delle scadenze e di quant'altro sia comodo visualizzare su un calendario.

4.6. Strumenti per il collaudo e la qualità

JUnit

Riferimento	http://www.junit.org/
Licenza	Common Public Licence
Descrizione	JUnit è un framework di testing per il linguaggio Java. Fornisce un modo semplice per testare in modo esplicito aree specifiche di un programma Java.
Motivazioni della scelta	Abbiamo scelto di utilizzare JUnit in quanto è comunemente considerata una delle migliori librerie Java mai sviluppate. L'utilizzo di questo framework forza la definizione di specifici risultati attesi nell'esecuzione di un dato programma, permettendo un controllo preciso della correttezza e

dell'attinenza di quanto sviluppato ai requisiti delineati: questo approccio di testing già canonizzato ci è sembrato una scelta migliore rispetto all'abituale testing generico e non strutturato.

Metrics 1.3.6

Riferimento	http://metrics.sourceforge.net/
Licenza	Eclipse Public Licence
Descrizione	Metrics è un plugin per Eclipse che permette di calcolare metriche e analizzare dipendenze. Permette di individuare cicli all'interno di pacchetti e dipendenze di tipo, e visualizzarli graficamente. Possono essere calcolate diverse metriche con deviazioni medie e standard.
Motivazioni della scelta	Non sono moltissimi gli strumenti gratuiti per il calcolo delle metriche: Metrics è un plugin per Eclipse quindi è stato semplice trovarlo, e si è rivelato semplice anche utilizzarlo. Metrics è altamente personalizzabile: i valori per le metriche, e anche parte della sua interfaccia, lo sono. Il suo essere integrato in Eclipse ci ha aiutato a svolgere le diverse operazioni tutte internamente al framework. I risultati dei calcoli effettuati da Metrics possono inoltre essere esportati in xml, permettendo una loro facile trasformazione in altri formati (pdf, xhtml).

Capitolo 5. Piano di qualità

Il presente piano intende stabilire l'insieme delle attività per la garanzia della qualità dell'intero progetto software. Il piano s'intende compreso e condiviso unanimemente da tutti i membri della squadra.

La guida di riferimento (<http://www.ambysoft.com/unifiedprocess/agileUP.html>) del modello di processo scelto suggerisce già delle attività necessarie per assicurare la qualità. Quest'ultime sono state opportunamente declinate nel contesto del presente progetto. Il piano di qualità infatti è stato stilato tenendo bene in mente quelli che sono i suoi costi ed il loro rapporto rispetto ai benefici. Considerate, quindi, le ridotte dimensioni e l'abbordabile complessità del progetto si è deciso di optare per attività di garanzia della qualità il più possibile snelle, in linea con il modello di processo scelto, cercando di non gravare eccessivamente sui costi.

Nelle prossime sezioni si definisce la portata del piano di qualità, le pratiche e gli strumenti a supporto, le responsabilità e le metriche utilizzate. Infine l'ultima sezione contiene il piano delle attività da svolgere per ogni fase e i risultati più significativi dei collaudi.

5.1. Portata del piano di qualità

Il controllo di qualità è applicabile a tutti gli artefatti prodotti lungo il ciclo di sviluppo del prodotto. L'elenco che segue definisce in questi termini la portata del presente piano:

- piano di processo;
- specifiche dei requisiti;
- manufatti della progettazione;
- software;
- documentazione.

5.2. Responsabili delle attività di garanzia della qualità

Considerata l'esiguità del numero di persone in seno al progetto e considerata la molteplicità di ruoli che le stesse devono ricoprire, la responsabilità delle attività per la garanzia della qualità (e.g. collaudi o revisioni) è spalmata in maniera più o meno uniforme tra i membri della squadra; tuttavia rimane inteso che il responsabile finale della qualità dei prodotti è il QE.

5.3. Pratiche, standard e strumenti a supporto della qualità

Le revisioni costituiscono lo strumento principale mediante il quale è possibile assicurare qualità. Esse costituiscono una sorta di filtro durante il processo di sviluppo e servono a scoprire errori e difetti di un qualsiasi artefatto. In letteratura si trovano due tipi di revisioni: quelle informali, condotte anche davanti ad una macchinetta del caffè tra i diretti interessati, e quelle formali, per le quali invece sono previste delle vere e proprie riunioni oltre che un dettagliato protocollo a cui attenersi. Per il presente progetto sono quindi pianificate due diversi tipi di revisioni: quelle informali, i cui esiti non saranno documentati, e una sorta di revisioni "formali". In particolare per le revisioni formali si stabilisce che:

- l'artefatto da revisionare sia revisionato da uno o più membri della squadra che non hanno partecipato alla sua realizzazione oppure dall'intera squadra;
- l'esito della revisione sia documentato in poche righe e sia notificato ai produttori dell'artefatto.

Le revisioni "formali" saranno comunque in numero ridotto e principalmente saranno fatte per abbattere i rischi più elevati derivanti dalla fase di progettazione.

Considerata la natura del modello di processo scelto si è deciso di adottare alcune pratiche proprie dell'Extreme Programming per ottenere i massimi benefici dal processo agile messo a punto. In particolare si stabilisce che i membri della squadra dovranno partecipare a brevissime riunioni mattutine, anche in piedi, utili per fare il punto della situazione, aggiornarsi sullo stato dei lavori e, perché no, rafforzare lo spirito di squadra. Un'altra pratica che dovrà essere adottata è quella della programmazione a coppie (c.d. pair programming): ciò dovrebbe consentire di ridurre al minimo il numero di bachi nel codice accorciando sensibilmente i tempi per il collaudo. In generale si consiglia un approccio Test Driven per la produzione di software: si dovranno sviluppare quanto prima test di regressione mediante l'utilizzo di JUnit.

Lo standard ISO 9126 identifica in sei attributi la qualità nel campo del software:

- funzionalità;
- affidabilità;
- facilità d'uso;
- efficienza;
- facilità di manutenzione;
- portabilità.

La qualità del software prodotto dovrà essere di volta in volta misurata rispetto a questi attributi. Per questi scopi si consiglia l'installazione del plugin EclipseMetrics per Eclipse al fine di monitorare la qualità del codice prodotto. EclipseMetrics calcola diverse metriche per il codice, è altamente configurabile e attraverso la vista dei problemi di Eclipse avverte di eventuali violazioni degli intervalli tollerati.

5.4. Metriche

Nella precedente sezione si è accennato all'utilizzo del plugin EclipseMetrics per il calcolo di diverse metriche per il codice sorgente. Quelle che effettivamente saranno calcolate e tenute in considerazione sono riassunte nell'elenco che segue:

- CC (Cyclomatic Complexity): misura il numero di sezioni di codice senza diramazioni all'interno di un metodo; è un indicatore della sua complessità;
- NOL (Number of Level): misura il livello di annidamento del metodo;
- NOS (Number of Statement): misura il numero di statement in un metodo;
- NOP (Number of Parameters): misura il numero di parametri di un metodo;
- NOF (Number of Field): misura il numero di campi di una classe;
- Ec (Efferent Coupling): misura l'accoppiamento di una classe;
- WMpC (Weighted Methods per Class): è la somma delle complessità ciclomeriche dei metodi della classe;

- LCOM-CK (Lack of Cohesion in Methods - Chidamber e Kemerer): misura la coesione di una classe secondo la metrica proposta da Chidamber e collega.

5.5. Attività per la garanzia della qualità

La garanzia di qualità si compone di diverse attività che attraversano tutto il processo software. La presente sezione contiene, per ogni fase, il piano dettagliato delle attività previste.

5.5.1. Durante tutte le fasi

Uno dei rischi maggiori connessi allo sviluppo del progetto è l'inesperienza dei membri della squadra; pertanto si stabilisce che il piano di processo debba essere continuamente monitorato, in maniera informale, durante tutte le fasi dello sviluppo. Eventuali divergenze significative dovranno essere prese in considerazione e discusse all'interno della squadra al fine di comprenderne le cause. Solo attraverso la comprensione di quest'ultime si sarà in grado di mantenere un adeguato controllo sul processo.

La qualità della presente relazione dovrà anch'essa essere monitorata durante tutto il ciclo di sviluppo. La responsabilità dei controlli è affidata a tutti i membri della squadra. La qualità della documentazione dovrà essere valutata in termini di completezza, chiarezza e correttezza. Si dovrà prestare attenzione anche alla qualità della formattazione dei contenuti nonché dell'impaginazione. I controlli, che manterranno un carattere informale, dovranno essere effettuati tramite opportune revisioni incrociate.

5.5.2. Fase di avvio

Revisione delle specifiche dei requisiti

Revisori: squadra.

Verifiche: completezza delle specifiche dei requisiti rispetto al documento dei requisiti, chiarezza, coerenza, tracciabilità, precisione, assenza di ambiguità nella comprensione tra i membri della squadra, convergenza sull'assegnazione delle priorità.

Esito: revisione effettuata in data 30/04/2009. Chiarite alcune divergenze sull'assegnazione delle priorità.

Revisione del modello dei casi d'uso

Revisori: squadra.

Verifiche: completezza e coerenza dei casi d'uso rispetto ai requisiti funzionali e viceversa, corretta identificazione degli attori, coerenza chiarezza e completezza nella descrizione degli scenari e dei flussi alternativi, completezza e correttezza delle relazioni tra casi d'uso.

Esito: revisione effettuata in data 30/04/2009. Ristrutturazione di parte del diagramma UML: eliminazione di qualche caso d'uso, individuazione di nuove relazioni tra casi d'uso. Chiarimenti sul flusso degli eventi.

Revisione del modello architetturale

Revisori: TS, PM.

Verifiche: complessità, coesione, completezza rispetto al problema, compatibilità con le specifiche dei requisiti, dimensioni, chiarezza.

Esito (TS): revisione effettuata in data 4/5/2009. Entrambi i modelli sono semplici e abbastanza astratti: il sistema che rappresentano non è più complesso del previsto, e di sicura fattibilità. I modelli sono compatibili coi requisiti, non propongono soluzioni non necessarie agli scopi del progetto, ma

si limitano a soddisfare alcuni dei requisiti di base. Non si rileva che vi siano requisiti che avrebbero dovuto essere espressi nel modello che siano stati tralasciati.

Esito (PM): revisione effettuata in data 6/5/2009. I diagrammi non rispettano del tutto i requisiti R1N e R2N, poiché non aderiscono allo schema a tre livelli consueto per le applicazioni realizzate con tecnologia J2EE (in particolare la comunicazione fra il livello informativo e il livello di interfaccia utente dovrebbe sempre essere veicolata attraverso il livello business).

Revisione del piano preliminare di schieramento

La revisione del piano preliminare di schieramento sarà informale e non necessiterà di essere documentata.

Revisione del prototipo dell'interfaccia

La revisione del prototipo dell'interfaccia sarà informale e non necessiterà di essere documentata.

5.5.3. Fase di elaborazione

Revisione del modello architetturale

Revisori: QE

Verifiche: complessità, coesione, completezza rispetto al problema, compatibilità con le specifiche dei requisiti, dimensioni, chiarezza, compatibilità rispetto al piano di schieramento, calcolo dell'indice di qualità strutturale DSQI.

Esito: il modello architetturale raffina coerentemente il modello elaborato e rivisto in fase di avvio. È pienamente compatibile con le specifiche dei requisiti e mantiene dimensioni più che accettabili. La strutturazione a quattro livelli tipica delle applicazioni enterprise conferisce al progetto un adeguato accoppiamento tra i moduli così come una buona coesione. Si rileva la compatibilità del modello rispetto al piano di schieramento.

Collaudo del prototipo architetturale

Revisori: squadra.

Verifiche: coerenza del prototipo rispetto al modello architetturale, collaudo delle interazioni tra i tre principali livelli architetturali: scrivere e visualizzare tramite browser dei dati memorizzati sul sistema informativo. Verifica della compatibilità del prototipo rispetto ai requisiti R1N, R2N, R3N.2, R4N.3 e R4N.1

Esito: il prototipo ha superato con successo le verifiche previste.

Revisione del piano di schieramento

La revisione del piano di schieramento sarà informale e non necessiterà di essere documentata.

5.5.4. Fase di costruzione

Collaudo delle unità

Collaudatori: squadra.

Modalità di collaudo: sviluppare test di regressione con JUnit; in particolare si consiglia di fare attenzione ai test sui valori limite e sui flussi eccezionali.

Collaudo di integrazione

Collaudatori: squadra.

Modalità di collaudo: verificare la corretta integrazione del modulo business, del modulo presentazione e del modulo informativo in un singolo nodo Jboss.

Collaudo dello schieramento

Collaudatori: squadra.

Modalità di collaudo: schierare l'applicazione in due cluster omogenei nelle macchine dipartimentali: verificare la corretta configurazione dei nodi JBoss, del database MySQL e del load balancer.

5.5.5. Fase di transizione

Validazione

Collaudatori: squadra.

Modalità di collaudo: verificare che l'applicazione soddisfi tutti i requisiti mediante user testing basato sui casi d'uso.

5.6. Risultati delle misurazioni e dei collaudi sul software

A titolo d'esempio si riporta in questa sezione un output del plugin EclipseMetrics relativo ai pacchetti g8.bookshop.business.core e g8.bookshop.business.util. Nella prima tabella sono presenti le metriche calcolate sulle classi, mentre nella seconda sui metodi.

Figura 5.1. Metriche sulle classi

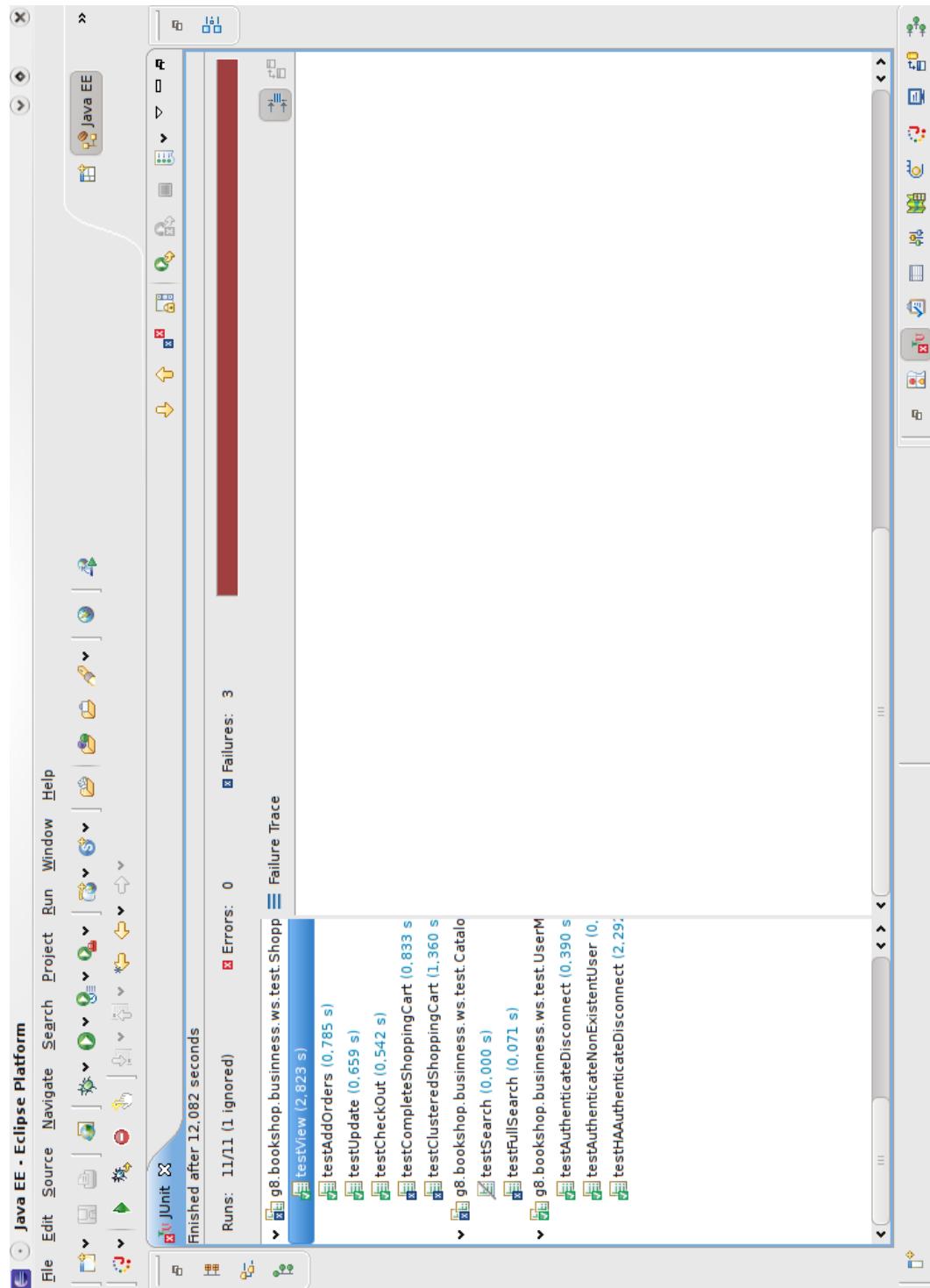
CC (max)	FE (max)	LOCm (max)	NLS (max)	NOL (max)	NOP (max)	NOS (max)	Ce	LCOM-CK	NOF	WMC	Line	Type	Package	
1	0	5	0	1	2	3	4	1	2	6	18	Order	g8.bookshop.business.core	
2	1	3	2	2	0	10	5	0	1	4	19	Customer	g8.bookshop.business.core	
2	2	5	3	2	1	13	10	0	1	8	23	ShoppingCart	g8.bookshop.business.core	
3	7	6	6	3	2	27	32	0	1	13	49	Converter	g8.bookshop.business.util	
-	-	-	-	-	0	-	-	-	-	-	6	CustomerRemote	g8.bookshop.business.core	
1	1	3	0	1	0	2	2	-	0	1	16	Guest	g8.bookshop.business.core	
-	-	-	-	-	1	-	-	-	-	-	8	OrderRemote	g8.bookshop.business.core	
-	-	-	-	-	1	-	-	-	-	-	12	ShoppingCartRemote	g8.bookshop.business.core	
1	0	4	0	1	1	1	4	-	3	3	16	User	g8.bookshop.business.core	
-	-	-	-	-	1	-	-	-	-	-	10	UserRemote	g8.bookshop.business.core	
1	-	12	4	1	3	13	6	-	0	2	18	BeanLocator	g8.bookshop.business.util	
-	-	-	-	-	1	-	-	-	-	-	19	ConverterRemote	g8.bookshop.business.util	
-	-	-	-	-	-	-	1	-	0	0	3	Name	g8.bookshop.business.util	
-	-	-	-	-	-	-	-	2	-	0	0	4	Name.EJB	g8.bookshop.business.util
-	-	-	-	-	-	-	-	2	-	0	0	37	Name.JMX	g8.bookshop.business.util

Figura 5.2. Metriche sui metodi

CC	FE	LOCm	NLS	NOL	NOP	NOS	Line	Method	Type	Package
3	0	0	6	3	1	16	162	xmIToOrders(java.lang.String)	Converter	g8.bookshop.business.util
2	0	0	2	2	0	10	39	createShoppingCart()	Customer	g8.bookshop.business.core
2	2	4	1	2	1	13	34	addOrder(g8.bookshop.business.core.OrderRemote)	ShoppingCart	g8.bookshop.business.core
2	0	4	3	2	1	9	48	addOrders(java.util.List)	ShoppingCart	g8.bookshop.business.core
2	0	0	2	2	1	12	77	booksToXML(java.util.List)	Converter	g8.bookshop.business.util
2	0	0	3	2	1	14	110	shoppingCartToXML(g8.bookshop.business.core.ShoppingCartRemote)	Converter	g8.bookshop.business.util
2	1	6	6	2	1	27	129	xmIToOrder(org.w3c.dom.Node)	Converter	g8.bookshop.business.util
1	1	3	0	1	0	2	31	Customer()	Customer	g8.bookshop.business.core
1	0	3	0	1	0	3	23	getShoppingCart()	Customer	g8.bookshop.business.core
1	1	3	0	1	0	2	18	Guest()	Guest	g8.bookshop.business.core
1	0	3	0	1	0	0	33	Order()	Order	g8.bookshop.business.core
1	0	5	0	1	2	3	22	Order(g8.bookshop.persistence.Book, int)	Order	g8.bookshop.business.core
1	0	3	0	1	0	1	39	getBook()	Order	g8.bookshop.business.core
1	0	3	0	1	0	1	45	getQuantity()	Order	g8.bookshop.business.core
1	0	5	0	1	1	1	60	setBook(g8.bookshop.persistence.Book)	Order	g8.bookshop.business.core
1	0	5	0	1	1	1	51	setQuantity(int)	Order	g8.bookshop.business.core
1	0	3	0	1	0	2	27	ShoppingCart()	ShoppingCart	g8.bookshop.business.core
1	0	4	0	1	0	2	73	checkOut()	ShoppingCart	g8.bookshop.business.core
1	0	4	0	1	0	2	82	getOrders()	ShoppingCart	g8.bookshop.business.core
1	0	5	0	1	1	4	63	update(java.util.List)	ShoppingCart	g8.bookshop.business.core
1	0	3	0	1	0	1	22	getId()	User	g8.bookshop.business.core
1	0	4	0	1	0	1	36	isCustomer()	User	g8.bookshop.business.core
1	0	3	0	1	1	1	29	setId(java.lang.String)	User	g8.bookshop.business.core
1	-	8	1	1	1	5	20	getBean(java.lang.String)	BeanLocator	g8.bookshop.business.util
1	-	12	4	1	3	13	34	getBean(java.lang.String, java.lang.String, java.lang.String)	BeanLocator	g8.bookshop.business.util
1	7	5	1	1	2	20	57	bookToXML(org.w3c.dom.Document, g8.bookshop.persistence.Book)	Converter	g8.bookshop.business.util
1	1	5	1	1	2	9	94	orderToXML(org.w3c.dom.Document, g8.bookshop.business.core.OrderRemote)	Converter	g8.bookshop.business.util
1	0	0	5	1	1	11	181	xmlDocumentCreator(java.lang.String)	Converter	g8.bookshop.business.util
1	0	0	4	1	1	10	194	xmlDocumentToString(org.w3c.dom.Document)	Converter	g8.bookshop.business.util

Nella figura che segue invece è stato catturato un output di JUnit durante il ciclo di sviluppo del prodotto.

Figura 5.3. Metriche sulle classi



Capitolo 6. Manuale utente

6.1. Panoramica

G8Bookshop è un negozio di libri online di semplice utilizzo. Le funzionalità di base che offre sono:

- ricerca di libri
- selezione di libri e inserimento in un carrello personale
- modifica del contenuto del carrello

In altre parole G8Bookshop permette di:

- cercare libri
- selezionare libri che desiderate acquistare e ricordarli mentre effettuate altre ricerche
- modificare i libri selezionati per l'acquisto rimuovendoli o modificando il numero di copie desiderato
- effettuare diverse ricerche e selezionare libri in momenti diversi
- acquistare i vostri libri!

6.2. Visitatori e clienti

L'acquisto di libri è possibile solo ad utenti in possesso di uno username e una password che abbiano effettuato la procedura detta "autenticazione", ovvero che abbiano inserito il loro username e password nei campi appropriati e siano stati riconosciuti dal sistema: questi utenti vengono detti *clienti*. Se non possedete una coppia username e password, o non l'avete fornita al sistema, siete invece utenti *visitatori*. Non è possibile allo stato attuale effettuare registrazioni presso G8Bookshop, data la natura sperimentale del sistema. Arrivando sulla pagina principale di G8Bookshop avrete di fronte due scelte possibili:

- effettuare ricerche mediante il form di ricerca centrale
- autenticarvi selezionando "Authenticate"

Come visitatori, potete ricercare liberamente libri all'interno del nostro catalogo, tuttavia non vi sarà possibile selezionarli tantomeno acquistarli. Se possedete i requisiti richiesti (una valida coppia username/password) potete effettuare la procedura di autenticazione: il sistema vi riconoscerà come clienti e potrete effettuare selezioni e acquisti. Potete effettuare tutte le ricerche che desiderate e autenticarvi in un secondo momento: la vostra ultima ricerca vi attenderà dopo l'autenticazione. Tuttavia, fino a che non vi sarete autenticati non vi sarà possibile selezionare libri. Se possedete uno username e una password proseguite la lettura alla sezione successiva per scoprire come acquistare libri.

Dopo aver effettuato la procedura di autenticazione, potrete effettuare in ogni momento la procedura opposta, la disconnessione, semplicemente selezionando "Disconnect". Ritornerete ad essere un semplice visitatore, e potrete effettuare ricerche senza selezionare libri, o autenticarvi con una diversa coppia username/password. Potete disconnettervi in qualunque momento: durante una ricerca, una modifica di carrello o una selezione. Non dimenticate di disconnettervi al termine della vostra navigazione su G8Bookshop!

6.3. Il carrello

Avete effettuato la procedura di autenticazione? Da questo momento in poi vi è possibile selezionare libri che desiderate acquistare. Il sistema vi fornisce, appena autenticati, un *carrello* virtuale nel quale

metterà i libri da voi selezionati. Potete visionarne in ogni momento il contenuto selezionando "View Cart". Appena autenticati il carrello vi apparirà vuoto: sarete voi a riempirlo selezionando libri.

Ora che siete dei *clienti* potete notare che i risultati della vostra ricerca vi appaiono in modo diverso: vi è per ognuno di essi un riquadro: se siete interessati a libri che risultano dalla vostra ricerca selezionate il riquadro associato e poi selezionate "Select", che troverete in fondo alla lista dei risultati. I libri scelti si troveranno nel vostro carrello: potete vederli selezionando "View Cart". Una volta visualizzato il contenuto del vostro carrello potete modificarlo: cancellate libri che non desiderate impostando la loro quantità a zero, e modificate la quantità dei libri che volete in più copie; al termine delle modifiche sul vostro carrello non dimenticate di selezionare "Update", o le modifiche non verranno effettuate. Dopo l'aggiornamento vi verrà mostrato il vostro nuovo carrello con le modifiche richieste. Potete procedere all'acquisto selezionando "Checkout", o selezionare "Continue to Search" se desiderate cercare altri libri: in questo caso i libri già inseriti nel carrello vi rimarranno, e i libri inseriti durante le nuove ricerche verranno aggiunti ai libri già presenti. Potete fare ricerche e selezionare libri, visualizzare il carrello e modificarne il contenuto, quante volte lo desiderate: i vostri libri verranno aggiunti al carrello man mano, fino a che non deciderete di procedere alla procedura definitiva di acquisto.

6.4. Acquisto

Potete acquistare il contenuto del vostro carrello semplicemente visualizzandolo e selezionando "Checkout". Se desiderate modificare il numero delle vostre copie un'ultima volta, potete farlo tranquillamente e poi selezionare direttamente "Checkout". G8Bookshop aggiornerà il carrello con le vostre ultime modifiche prima di procedere all'acquisto. Fate attenzione: solo i libri presenti nel carrello, col numero di copie ivi specificato, verranno acquistati. Non è possibile acquistare libri non presenti nel carrello, nè è possibile non acquistare un libro che ci trova nel vostro carrello. Selezionate ogni libro che desiderate acquistare e controllate che il numero di copie per libro che risulta nel vostro carrello sia quello desiderato, e accertatevi che il vostro carrello non contenga libri che non volete acquistare, prima di procedere al "Checkout".

Dopo l'acquisto il contenuto del vostro carrello vi sarà mostrato nuovamente, così potrete revisionare i libri comprati. Potete ignorarlo e proseguire con le ricerche per riempire di nuovo il carrello oppure disconnettervi e abbandonare G8Bookshop.

Capitolo 7. Valutazione dello sforzo

7.1. Valutazione preventiva

La valutazione preventiva dello sforzo è stata utile, sin dalle prime fasi del progetto per stimare i tempi ad esso necessario e prevederne così la conclusione. La valutazione effettuata, fondata sulla metrica dei punti funzione, è riportata nella presente sezione.

7.1.1. Calcolo dei punti funzione

Il calcolo della metrica è stato effettuato secondo il procedimento più classico, proposto da Albrecht. La tabella che segue è mostrato il computo dei punti funzione ancora non aggiustati, comprensiva di qualche commento ai valori scelti.

Tabella 7.1. Calcolo punti funzione non aggiustati

Fattore	Quantità	Complessità	Totale
Input utente	3 ricerca, credenziali, gestione carrello	4 media	12
Output utente	2 ricerca, carrello	7 media	14
Richieste utente	0		
File logici	2 libro, credenziali	10 media	20
Interfacce esterne	0		
Conto totale grezzo			35

L'aggiustamento è stato poi calcolato attraverso il noto questionario riportato di seguito:

1. Il sistema ha bisogno di operazioni di backup e ripristino affidabili? [0]
2. È necessario impiegare comunicazioni specializzate per trasferire le informazioni da o verso l'applicazione? [5]
3. Esistono funzioni di elaborazione distribuita? [3]
4. Le prestazioni rappresentano un fattore critico? [1]
5. Il sistema può funzionare in un ambiente operativo esistente, pesantemente utilizzato? [5]
6. Il sistema richiede un inserimento online dei dati? [1]
7. L'inserimento online dei dati richiede che venga realizzata una transazione di input costituita da più schermi od operazioni? [0]
8. I file IFL vengono aggiornati online? [0]
9. Esistono input, output, file o richieste di natura complessa? [1]
10. L'elaborazione interna è complessa? [2]
11. Il codice è progettato per essere riutilizzabile? [5]

12.Nel progetto sono compresi la conversione e l'installazione? [4]

13.Il sistema è progettato per più installazioni in più organizzazioni? [4]

14.L'applicazione è progettata per facilitare le modifiche e la facilità d'uso da parte degli utenti? [3]

Il risultato totale di 34 costituisce il valore di aggiustamento da sostituire nell'equazione empirica assieme al conto totale grezzo precedentemente ottenuto.

Punti funzione = "conto totale grezzo" x (0.65 + 0.01 x "valore di aggiustamento")

Si ottiene quindi il numero di punti funzione dell'applicazione.

Punti funzione = 35 x (0.65 + 0.01 x 34) = 34.63

7.1.2. Stima dello sforzo

Per avvicinare la metrica dei punti funzione ad una quantificazione dello sforzo concreta e verificabile si è anzitutto deciso di stimare le linee di codice sorgente che saranno prodotte. Sfruttando una stima empirica del rapporto fra linee di codice di un software e i punti funzione in esso contenuti, che per applicazioni Java Enterprise, è pari a 50, si è quindi stimato che venissero prodotte 2114 righe di codice sorgente.

In secondo luogo si è voluto stimare lo sforzo anche secondo un'altra misura altamente verificabile: i giorni di lavoro per uomo. A partire da una valutazione empirica della produttività degli sviluppatori in una media di 15 punti funzione mensili, si è ricavato che lo sforzo globale previsto per il progetto è pari a 2,3 mesi/uomo. Per la squadra di quattro componenti tale stima equivale a poco più di 12 giorni di lavoro a testa.

7.2. Valutazione consuntiva

La valutazione consuntiva degli sforzi effettuati è stata elaborata sia per la stima di righe di codice che per la stima di ore di lavoro. Infatti, nonostante entrambe le stime siano basate sulla metrica dei punti funzione, è bene ricordare che, delle due, la prima stima è stata usata per la valutazione della seconda.

La quantità di codice sorgente prodotto dalla squadra è stato calcolato a partire dai codici Java Enterprise e dalle configurazioni dell'application server in XML. Sono stati inseriti nel conteggio i soli sorgenti effettivamente prodotti dalla squadra per l'applicazione, escludendo i sorgenti generati automaticamente da alcuni strumenti di sviluppo e tralasciando anche i codici usati per i collaudi di unità. Il conto totale di righe di codice ammonta a 2069 ed è davvero molto vicino alla previsione precedentemente illustrata.

Il conto delle ore di lavoro effettivamente spese dalla squadra è stato computato ancor più facilmente a partire dai diari che hanno puntualmente registrato le attività. Il diario tenuto da ogni componente e il diario degli incontri di gruppo, infatti, è stato stilato riportando le ore di sforzo dedicate ai lavori. La tabella seguente riassume la somma degli sforzi parziali dei componenti del team.

Tabella 7.2. Valutazione consuntiva dello sforzo

Componente	Ruolo	Sforzo (ore di lavoro)
Francesco Draicchio	Librarian	117
Andrea Rappini	Quality engineer	195.33
Silvia Righini	Tool specialist	127
Marco Solieri	Project manager	196.75
Squadra		40.5
Totale		675.58

Lo sforzo consuntivo espresso in mesi uomo supera di pochissimo i 4 mesi/uomo, equivalenti a circa 21,1 giorni per componente. Lo scarto fra la stima preventiva e la valutazione consuntiva costituisce un errore relativo del 74%, che supera di gran lunga la peggiore stima d'errore preventivata informalmente nelle fasi di pianificazione del progetto.

La causa prima di questa discrepanza risiede sicuramente nell'inesperienza gestionale e attuativa nello specifico ambito di progetto di cui è parlato nella sezione 3.4. Non si può trascurare, peraltro, che in accordo con la legge di Parkinson, le date di consegna multiple abbiano aumentato le necessità di sforzo per il progetto, dal momento che hanno creato disponibilità di sforzo.

Capitolo 8. Diari

8.1. Diario di Gruppo

Giorno	Quantità di ore	Descrizione
06/04/2009	2	<ul style="list-style-type: none">• Scelta del nome del gruppo• Divisione dei ruoli
07/04/2009	3	<ul style="list-style-type: none">• Definizione del piano di processo.• Prima bozza di definizione dell'Environment.
15/04/2009	3	<ul style="list-style-type: none">• Prima stesura dei requisiti.
20/04/2009	1.5	<ul style="list-style-type: none">• Decisioni su specifici punti del piano di processo.• Punto della situazione attuale.• Divisione di compiti relativi alla fase di avvio.
27/04/2009	2.5	<ul style="list-style-type: none">• Condivisione di alcune conoscenze acquisite in maniera individuale (principalmente su JBoss).• Notifica di alcuni problemi relativi all'uso di Docbook e loro risoluzione.• Analisi collettiva di una prima bozza di casi d'uso e chiarimenti sugli attori coinvolti.• Modifiche terminologiche e aggiornamento del glossario.
30/04/2009	2.5	<ul style="list-style-type: none">• Revisione piano di progetto: definizione prossime scadenze, pianificazione prossime attività.• Revisione dell'architettura.• Revisione casi d'uso.• Aggiornamenti vari.
5/05/2009	1.5	<ul style="list-style-type: none">• Prototipazione tecnica.
7/05/2009	3.5	<ul style="list-style-type: none">• Prototipazione tecnica.
12/05/2009	3.5	<ul style="list-style-type: none">• Prototipazione tecnica.
18/05/2009	3	Prototipazione tecnica.
19/05/2009	2.5	<ul style="list-style-type: none">• Prototipazione tecnica.
21/05/2009	1.5	<ul style="list-style-type: none">• Prototipazione tecnica.
25/05/2009	2	<ul style="list-style-type: none">• Prototipazione tecnica.
27/05/2009	5	<ul style="list-style-type: none">• Modellazione delle componenti.• Modellazione logica di business.
03/06/2009	3	<ul style="list-style-type: none">• Prototipazione tecnica.
04/06/2009	8	<ul style="list-style-type: none">• Elaborazione.• Revisione informale manufatti modellazione logica business.
05/06/2009	8	<ul style="list-style-type: none">• Elaborazione.
Totale	40.5 ore	Fino al 5/6/2009

Giorno	Quantità di ore	Descrizione
		<ul style="list-style-type: none">• Revisione informale manufatti modellazione logica business.
Totale	40.5 ore	Fino al 5/6/2009

A partire dall'ultima data indicata in tabella i componenti del gruppo si sono trovati regolarmente quasi ogni giorno: gli incontri di gruppo non stati quindi registrati come tali in quanto si sono trasformati in brevi momenti di revisioni e aggiornamenti informali durante il lavoro individuale.

8.2. Diario di Marco Solieri

Tabella 8.1. Diario del Project Manager

Data	Durata (ore)	Descrizione
07/04/2009	1.5	Approfondimenti di gestione di progetti. Analisi modelli di processo agili (XP, AUP), scelta e documentazione approfondita
15/04/2009	1.5	Approfondimenti su J2EE e JBoss. Analisi delle specifiche e modellazione preliminare dei requisiti.
16/04/2009	2	Approfondimento di AUP, definizione preliminare del piano di processo (fase di avvio).
16/04/2009	1.3	Impostazione deposito versionato (SVN), aggiornamento wiki, programmazione prossimo incontro
16/04/2009	0.7	Approfondimento sulle metriche di valutazione dello sforzo (FP)
17/04/2009	3	Approfondimento di AUP, definizione preliminare piano di processo (fasi di elaborazione e costruzione)
20/04/2009	3	Definizione preliminare piano di processo (fase di transizione). Analisi strumenti per la gestione di progetti (scelto KPlato), definizione dettagliata piano di processo della fase corrente.
21/04/2009	1	Analisi di strumenti di elaborazione DocBook (OpenOffice), aggiornamento del presente Diario.
22/04/2009	3.5	Studio strumenti di gestione progetto alternativi e più confacenti le esigenze, apprendimento d'uso di TaskJuggler. Analisi dipendenze delle attività della fase Avvio.
27/04/2009	1	Analisi requisiti e relative priorità e aggiornamento del glossario (col QE). Tentativi d'utilizzo di DBLaTeX.
27/04/2009	1.5	Aggiornamento del documento dei requisiti e del glossario. Installazione xsltproc e trasformatore da DB a XHTML e apprendimento d'uso.
28/04/2009	2.5	Impostazione preliminare del piano di progetto (e apprendimento d'uso di TaskJuggler).
28/04/2009	1	Aggiornamento del progresso del progetto (e apprendimento d'uso di TaskJuggler).
29/04/2009	1	Revisione incrociata con il QE della prima versione del piano preliminare di processo e del piano preliminare di qualità.
29/04/2009	1	Reperimento e installazione di Eclipse EE IDE, del relativo plugin per l'integrazione SVN e di JBoss AS. Prime confidenze e prime configurazioni degli strumenti.
29/04/2009	2	Formazione sulla gestione di progetto in generale e nel modello AUP. Ampliamento del preliminare alla fase di elaborazione.
05/05/2009	1	Chiusura fase di avvio, condivisione artefatti di processo: riordinamenti nel deposito di versionamento, condivisione dei prototipi implementativi, del calendario e del rapporto di stato del processo.
06/05/2009	0.5	Revisione modellazione dell'architettura.
07/05/2009	0.5	Modellazione preliminare dell'architettura: correzione diagramma.
07/05/2009	1	Revisione modelli: revisione del prototipo di interfaccia utente (con QE).
Totale	196.75	Aggiornato al 04/07/2009

Data	Durata (ore)	Descrizione
07/05/2009	0.5	Gestione progetto: aggiornamento dello stato di avanzamento e dello sforzo consuntivo.
08/05/2009	2	Formazione architetturale e implementativa: studio tecnologia J2EE e prove con JBoss AS.
11/05/2009	1	Formazione architetturale e implementativa: studio J2EE e JBoss AS.
11/05/2009	4	Prototipazione architetturale: prime bozze implementative e prove di schieramento su CS.
11/05/2009	0.5	Configurazione ambiente: aggiornamento di JBOSS AS e installazione plugin JBOSS Tools.
12/05/2009	0.5	Formazione architetturale e implementativa: studio J2EE e JBoss AS.
12/05/2009	1.5	Prototipazione architetturale: esposizione di un EJB tramite WS.
13/05/2009	1.5	Prototipazione architetturale: generazione di un end-point di un WS.
14/05/2009	1.5	Prototipazione architetturale: collegamento di una servlet ad un endpoint di un WS.
14/05/2009	0.5	Condivisione artefatti: aggiornamento diario personale.
21/05/2009	1.5	Prototipazione architetturale. Formazione uso entity bean e basi di dati.
24/05/2009	2	Modellazione architetturale definitiva. Modellazione schieramento.
28/05/2009	3	Modellazione livello business: interfaccia ws e microarchitettura del livello (con QE).
29/05/2009	1	Modellazione livello business: microarchitettura del livello (con QE).
01/06/2009	2	Implementazione livello business: ws tramite stateless ejb.
03/06/2009	3	Modellazione livello business: microarchitettura (con QE).
03/06/2009	3	Modellazione business: linguaggio XML per i ws (con TS).
03/06/2009	1.5	Configurazione ambiente: risoluzione problemi legati a Subversion nella la macchina di sviluppo.
03/06/2009	1	Formazione: approfondimenti sull'uso delle Entity.
03/06/2009	2.5	Modellazione ed implementazione business.
04/06/2009	1	Configurazione per schieramento: impostazione DBMS MySQL e integrazione con Eclipse (con QE).
04/06/2009	2	Formazione: approfondimenti sull'uso DBMS MySQL (con QE).
04/06/2009	2	Modellazione ed implementazione business: attività nel lato informativo.
04/06/2009	1	Configurazione schieramento: impostazione data source di JBoss.
04/06/2009	0.5	Condivisione artefatti: condivisione progetti e impostazioni AS
05/06/2009	0.5	Aggiornamento diario personale
05/06/2009	3	Implementazione e formazione: business core - problemi con UserManager
08/06/2009	3	Modellazione business (col QE)
09/06/2009	1	Implementazione ws business: bozze di uso del modulo core
09/06/2009	4	Formazione: apprendimento estensione EJB3 di JBoss (col QE)
09/06/2009	3	Implementazione business: costruzione user manager (quasi completa)
10/06/2009	5	Implementazione business: costruzione shopping cart (inizio) (col QE))
Totale	196.75	Aggiornato al 04/07/2009

Data	Durata (ore)	Descrizione
10/06/2009	1.5	Implementazione business: costruzione shopping cart (conclusione) (col QE).
11/06/2009	0.25	Aggiornamento del diario personale.
11/06/2009	5.5	Correzioni e collaudi al business core (con QE).
12/06/2009	3	Correzioni e collaudi al business core (con QE e L).
13/06/2009	3	Formazione sul clustering di JBossAS.
15/06/2009	9	Clusterizzazione del business: collaudi (con QE).
16/06/2009	9.5	Clusterizzazione del business: collaudi (con QE).
17/06/2009	10.5	Clusterizzazione del business: collaudi (con QE).
18/06/2009	9	Clusterizzazione del business: collaudi (con QE).
19/06/2009	1	Incontro con lo stakeholder Prof.ssa Giorgia Lodi.
19/06/2009	4	Clusterizzazione del business: collaudi (con QE).
22/06/2009	2	Clusterizzazione del business: collaudi (con QE).
23/06/2009	7.5	Clusterizzazione del business: collaudi (con QE).
24/06/2009	0.5	Aggiornamento diario personale.
24/06/2009	7	Clusterizzazione del business: collaudi (con QE).
27/06/2009	2	Clusterizzazione del business: collaudi e correzioni.
29/06/2009	1	Bozza nuovi diagrammi business.
30/06/2009	1	Bozza nuovi diagrammi business.
30/06/2009	3	Schieramento sulle macchine dipartimentali: collaudi (con QE).
01/07/2009	4	Bozza dei contenuti della relazione: stima dello sforzo tramite punti funzione (PF).
2/07/2009	3	Documentazione: bozza relazione di progetto del livello business.
2/07/2009	2	Rifattorizzazione del livello business.
3/07/2009	7	Documentazione: raffinamento diagrammi del livello business.
4/07/2009	2	Bozza capitolo della documentazione di progetto (con QE): relazione di progetto
4/07/2009	5	Stesura capitoli della documentazione di progetto: piano di processo e valutazioni dello sforzo
Totale	196.75	Aggiornato al 04/07/2009

8.3. Diario di Andrea Rappini

Tabella 8.2. Diario del Quality Engineer

Giorno	Quantità di ore	Descrizione
5/4/09	4	Studio delle attività per SQA.
8/4/09	1.5	Studio delle attività per SQA.
9/4/09	3	Studio delle attività per SQA e bozza.
10/4/09	3	Analisi dei requisiti e studio di JBOSS.
14/4/09	1	Studio di JBOSS.
16/4/09	4.25	Con PM: definizione preliminare del processo e valutazione preliminare dello sforzo (metodo dei punti funzione).
21/4/09	1	Valutazione di strumenti per la scrittura di documenti DocBook.
22/4/09	1	Gestione del processo (con PM).
25/4/09	1	Studio documentazione su JBOSS.
25/4/09	0.5	Prima stesura specifiche dei requisiti.
25/4/09	0.5	Bozza di piano di qualità.
26/4/09	3	Valutazione di strumenti per il collaudo.
27/4/09	1	Con il PM: priorità dei requisiti
27/4/09	0.5	Con il PM: gestione del progetto
28/4/09	4	Studio di JBOSS: realizzazione di prototipi minimali.
29/4/09	1	Revisione incrociata con il PM della prima versione del piano preliminare di processo e del piano preliminare di qualità.
30/4/09	2	Revisione con il PM del piano preliminare di qualità e prima integrazione nel piano di processo.
1/5/09	0.5	Aggiornamento diario.
1/5/09	0.5	Installazione e collaudo di strumenti per la conversione di documenti DocBook.
1/5/09	1	Prima stesura e versionamento del piano di qualità.
2/5/09	1.75	Approfondimento delle attività di SQA. Aggiornamento ed estensione del piano di qualità.
3/5/09	2.5	Installazione e apprendimento di Eclipse TPTP per i collaudi d'unità e integrazione.
4/5/09	2	Realizzazione prototipo interfaccia.
5/5/09	1	Formazione su Eclipse TPTP.
7/5/09	0.5	Revisione del modello architettonico preliminare (con PM).
7/5/09	1	Revisione del prototipo dell'interfaccia (con PM).
8/5/09	1.25	Formazione sui Web Services.
11/5/09	4.5	Realizzazione di prototipi minimali con JBoss e prove di schieramento su CS.
12/5/09	1	Prototipi architettonici: esposizione di una EJB come Web Service.
13/5/09	2.5	Prototipi architettonici: esposizione di una EJB come Web Service e creazione del relativo client.
Totale	215.33	Aggiornato al 5/7/2009

Giorno	Quantità di ore	Descrizione
15/5/09	1	Prototipi architetturali: esposizione di una EJB come Web Service e creazione del relativo client.
15/5/09	0.5	Aggiornamento del diario personale.
24/5/09	1.5	Configurazione strumenti di sviluppo.
28/5/09	3	Progettazione del livello business (con PM).
29/5/09	1	Progettazione del livello business (con PM).
30/5/09	1	Progettazione del livello business.
3/6/09	7.33	Progettazione e prima implementazione del livello business.
5/6/09	3	Realizzazione prototipi per il livello informativo (con PM).
5/6/09	3	Progettazione del livello business e analisi delle specifiche J2EE.
8/6/09	6.5	Progettazione del livello business: servizi singleton.
9/6/09	6	Progettazione del livello business.
10/6/09	10.50	Implementazione del livello business e informativo.
11/6/09	9	Implementazione e primi collaudi di integrazione.
12/6/09	9	Implementazione e collaudi di integrazione.
13/6/09	3	Implementazione e sviluppo di casi collaudo.
14/6/09	4	Collaudi e aggiornamento del piano di qualità.
15/06/2009	9	Clusterizzazione del business: collaudi (con PM).
16/06/2009	9.5	Clusterizzazione del business: collaudi (con PM).
17/06/2009	10.5	Clusterizzazione del business: collaudi (con PM).
18/06/2009	9	Clusterizzazione del business: collaudi (con PM).
19/06/2009	1	Incontro con lo stakeholder Prof.ssa Giorgia Lodi.
19/06/2009	4	Clusterizzazione del business: collaudi (con PM).
20/6/09	3	Implementazione: prove di schieramento e debug.
22/06/2009	2	Clusterizzazione del business: collaudi (con PM).
23/06/2009	7.5	Clusterizzazione del business: collaudi (con PM).
27/6/09	4	Implementazione: prove di schieramento e debug.
29/6/09	4	Implementazione: prove di schieramento e debug.
30/6/09	4	Implementazione: modifiche al business e prove di chieramento.
1/7/09	4	Documentazione: esteso il piano della qualità.
2/7/09	3	Business: refattorizzazione del livello business (con PM).
2/7/09	2	Documentazione: bozza relazione di progetto (con PM).
3/7/09	7	Documentazione: raffinamento diagrammi business (con PM).
4/7/09	2	Documentazione: bozza documentazione del business (con PM).
4/7/09	4	Documentazione: stesura documentazione del livello business.
5/7/09	4	Documentazione: raffinamenti al piano di qualità.
Totale	215.33	Aggiornato al 5/7/2009

8.4. Diario di Silvia Righini

Giorno	Quantità di ore	Descrizione
10/04/2009	6	<ul style="list-style-type: none"> Analisi di Docbook e degli strumenti per un suo utilizzo. Prima stesura della bozza di relazione. Preparazione tabelle per il mantenimento dei diari.
15/04/2009	3	<ul style="list-style-type: none"> Sviluppo di alcuni fogli di stile specifici per il mantenimento coerente della documentazione sul wiki e in forma di pdf. Creazione di tabella preliminare per i requisiti.
16/04/2009	4	<ul style="list-style-type: none"> Modifiche e tentativi di bugfix nel processo di generazione automatico dei diari. Studio sui casi d'uso e i diagrammi UML.
19/04/2009	4	<ul style="list-style-type: none"> Bugfix su generazione automatica diari. Configurazione di alcuni strumenti tra cui il client svn e pluging Eclipse. Primi passi con ArgoUML. Studio su UML.
22/04/2009	2	<ul style="list-style-type: none"> Prove con JBoss.
24/04/2009	3	<ul style="list-style-type: none"> Prove con JBoss. Prima bozza documento relativo all'environment.
26/04/2009	4	<ul style="list-style-type: none"> Prima bozza casi d'uso.
27/04/2009	2	<ul style="list-style-type: none"> Documentazione su J2EE e web services.
4/05/2009	3	<ul style="list-style-type: none"> Formalizzazione documentazione prodotta nell'incontro di gruppo precedente. Versione rivista dei casi d'uso: tabelle riviste, nuovo diagramma UML. Aggiornamento tabella di mappatura dei requisiti.
5/05/2009	2	<ul style="list-style-type: none"> Revisione della bozza della relazione dal punto di vista grafico. Commento/revisione al diagramma preliminare dell'architettura.
11/05/2009	2	<ul style="list-style-type: none"> prove con JBoss (EntityBean)
13/05/2009	4	<ul style="list-style-type: none"> prove con JBoss (comunicazione Servlet/SessionBean) Tentativo di fix problema relativo a Java 6 su Mac.
14/05/2009	2	<ul style="list-style-type: none"> prove con JBoss (comunicazione Servlet/SessionBean) Fix problema relativo a Java 6 su Mac.
20/05/2009	1	<ul style="list-style-type: none"> prove con JBoss.
25/05/2009	2	<ul style="list-style-type: none"> Documentazione ed elaborazione: JSP e servlet.
26/05/2009	4	<ul style="list-style-type: none"> Documentazione ed elaborazione: JSP e servlet.
29/05/2009	4	<ul style="list-style-type: none"> Documentazione ed elaborazione: JSP e servlet, JSTL.
Totale	127 ore	Fino al 3/7/2009

Giorno	Quantità di ore	Descrizione
		<ul style="list-style-type: none"> • Installazione dblatex.
6/06/2009	2	<ul style="list-style-type: none"> • Documentazione ed elaborazione: CSS e JSPF.
8/06/2009	4	<ul style="list-style-type: none"> • Elaborazione: JSP, servlet. • Documentazione su trasformazioni XSLT. • Aggiornamento diario.
9/06/2009	6	<ul style="list-style-type: none"> • Elaborazione: JSP, servlet. • Elaborazione: prove di trasformazioni XSLT, preparazione librerie e funzioni necessarie e di ausilio.
10/06/2009	3	<ul style="list-style-type: none"> • Elaborazione: fogli di stile XSLT per la parte presentazionale. • Elaborazione: css e JSP.
11/06/2009	6	<ul style="list-style-type: none"> • Controllo funzionamento effettivo di dblatex e modifica file della relazione per output ottimale. • CSS. • Fix al foglio di stile xslt.
12/06/2009	4	<ul style="list-style-type: none"> • Modifica immagini per inserimento nella relazione, composizione relazione.
13/06/2009	2	<ul style="list-style-type: none"> • Aggiornamento glossario ed environment.
15/06/2009	8	<ul style="list-style-type: none"> • Bugfix ed elaborazione: XSLT, JSP, CSS • Relazione: bugfix per uso con dblatex e per stampa ottimale con entrambi i trasformatori.
16/06/2009	4	<ul style="list-style-type: none"> • Conversione immagini per inserimento in relazione. • Modifica di configurazioni JBoss e MySql. • Refactoring relazione.
18/06/2009	3	<ul style="list-style-type: none"> • Refactoring relazione • Modifiche al foglio di stile.
24/06/2009	5	<ul style="list-style-type: none"> • Prove clusterizzazione e schieramento.
26/06/2009	5	<ul style="list-style-type: none"> • Prove clusterizzazione e schieramento. • Bugfix relazione.
29/06/2009	6	<ul style="list-style-type: none"> • Diagrammi UML: classi, package. • Prove sullo schieramento.
1/07/2009	5	<ul style="list-style-type: none"> • Diagrammi UML: classi, package. • Stesura parte presentation relazione.
2/07/2009	4	<ul style="list-style-type: none"> • Diagrammi UML: sequence parte presentation, deployment • Stesura parte presentation relazione. • Modifiche alla formattazione della relazione finale.
Totale	127 ore	Fino al 3/7/2009

Giorno	Quantità di ore	Descrizione
3/07/2009	8	<ul style="list-style-type: none">• Diagrammi UML: correzioni e disegno.• Correzioni/modifiche capitolo sugli strumenti.• Correzioni e refactoring sulla relazione.
4/07/2009	3	<ul style="list-style-type: none">• Diagrammi UML: correzioni e disegno.• Correzioni/modifiche/stesura relazione• Testing
Totale	127 ore	Fino al 3/7/2009

8.5. Diario di Francesco Draicchio

Giorno	Durata (ore)	Descrizione
08/04/2009	3	<ul style="list-style-type: none"> • aggiornamento del wiki di gruppo • trascrizione degli appunti formulati durante la seduta del 07/04 • studio della documentazione tecnica JBOSS e delle relative tecnologie
16/04/2009	3	<ul style="list-style-type: none"> • progettazione del sistema di formattazione automatica dei diari • realizzazione (proof of concept) del sistema di formattazione automatica dei diari (sincronizzazione repository SVN)
17/04/2009	2	<ul style="list-style-type: none"> • rielaborazione dello stylesheet di formattazione dei diari • bugfix motore di traduzione
21/04/2009	3	<ul style="list-style-type: none"> • elaborazione dei diagrammi a blocchi rappresentanti la struttura dei componenti dell'applicazione e modello dell'architettura client-server (livello di descrizione: alto) • studio e inizializzazione dell'ambiente di sviluppo composto da eclipse e jboss tools • inizializzazione del run-time enviroment: JBoss Server
23/04/2009	4	<ul style="list-style-type: none"> • PROOF OF CONCEPT: elaborazione di un webservice minimale (bean: HelloService) • studio del metodo di deploy dell'applicazione, generazione del deploy-descriptor e wsdl
25/04/2009	2.5	<ul style="list-style-type: none"> • elaborazione dell'applicazione client per verificare il funzionamento e l'effettivo deploy del webservice Hello • PROOF OF CONCEPT: studio ed elaborazione metodi di comunicazione tra applicazione e DBMS predefinito Hypersonic
27/04/2009	1.5	<ul style="list-style-type: none"> • BUGFIX: correzione di un problema riguardante il motore di trasformazione docbook->html: valutazione xi:xinclude, generalizzazione del processo di trasformazione
08/05/2009	2	<ul style="list-style-type: none"> • prove di trasformazione della documentazione da formato DocBook a Latex: rilevati alcuni problemi nella codifica
11/05/2009	3.5	<ul style="list-style-type: none"> • esercitazione JBoss: comunicazione tra componenti: in modo particolare tra Servlet e SessionBean tramite WebService
12/05/2009	2	<ul style="list-style-type: none"> • prove di comunicazione tra componenti: WebService e SessionBean
13/05/2009	4	<ul style="list-style-type: none"> • realizzazione architettura di comunicazione: bw -> servlet -> ws -> sessionbean
15/05/2009	1.5	<ul style="list-style-type: none"> • prove di comunicazione tra la logica di business e la parte informativa • inizializzazione DB di prova e prove di persistenza
19/05/2009	2	<ul style="list-style-type: none"> • test sulla persistenza del DataSource di default • interrogazione DB di prova
21/05/2009	6.5	<ul style="list-style-type: none"> • studio e prove tecniche di creazione di un nuovo data source mediante HSQL e relativa factory
Totale	117	Aggiornato al 04/07/2009

Giorno	Durata (ore)	Descrizione
		<ul style="list-style-type: none"> • studio e sperimentazione delle proprietà di persistenza
22/05/2009	3	<ul style="list-style-type: none"> • studio modalità di interrogazione del DB mediante EntityManager • creazione prototipale dell'architettura comprensiva della parte informativa persistente
28/05/2009	6	<ul style="list-style-type: none"> • sperimentazioni con session bean stateful
01/06/2009	1.5	<ul style="list-style-type: none"> • sperimentazioni con session bean stateful
05/06/2009	3	<ul style="list-style-type: none"> • refactoring nomi G8Presentation e prima impostazione del progetto Eclipse
07/06/2009	4.5	<ul style="list-style-type: none"> • prima implementazione della servlet Search
08/06/2009	4	<ul style="list-style-type: none"> • restyling delle pagine JSP mediante modifica dei moduli JSFPF • impostazione delle rimanenti servlet: Authenticate, Disconnect, View, Update, CheckOut
10/06/2009	4	<ul style="list-style-type: none"> • completamento definizione dell'architettura della logica di presentazione • definizione POJO DataExchange utilizzato come tramite tra servlet e JSP
11/06/2009	5.5	<ul style="list-style-type: none"> • primo test di comunicazione tra business e client mediante Search servlet e Catalogue Service
12/06/2009	4	<ul style="list-style-type: none"> • refactoring WSDL dei webservice e conseguente rigenerazione dei client lato Presentation • utilizzo delle librerie Axis client
13/06/2009	4.5	<ul style="list-style-type: none"> • implementazione Converter lato business: from Entity to XML, from XML to Entity
15/06/2009	6	<ul style="list-style-type: none"> • bug fixing presentazione sui file formatter.xsl, DataExchange, XsltTransformer • realizzazione Dispatcher delle richieste
17/06/2009	5	<ul style="list-style-type: none"> • modifica foglio di stile css, correzione formatter.xsl • refactoring definitivo nomi e funzionalità dei componenti lato presentation
19/06/2009	5.5	<ul style="list-style-type: none"> • bug fixing e collaudo presentazione • prime prove di clustering e load balancing
23/06/2009	4	<ul style="list-style-type: none"> • bug fixing e collaudo presentazione
24/06/2009	3.5	<ul style="list-style-type: none"> • schieramento a stato condiviso
26/06/2009	4	<ul style="list-style-type: none"> • collaudo clusterizzazione e prove di integrazione presentation/business, test tolleranza ai guasti lato presentazione
30/06/2009	4	<ul style="list-style-type: none"> • collaudo e correzione bachi dell'applicazione, rifiniti controlli e gestione delle eccezioni
02/07/2009	6	<ul style="list-style-type: none"> • piccole rifattorizzazioni, rifinita interfaccia, aggiornamento diario e screenshot applicazione
Totale	117	Aggiornato al 04/07/2009

Glossario

C

Carrello	Insieme di libri selezionati per l'acquisto.
Cassa (attività)	Attività che raggruppa le operazioni di visualizzazione del contenuto del carrello (VisualizzaCarrello), di modifica del contenuto (ModificaCarrello), e di saldo finale, detto anche acquisto definitivo od operazione di acquista carrello (AcquistaCarrello).
Catalogo	Insieme delle informazioni sui libri in vendita: titolo, autore, ISBN, editore, anno.
Cliente	Utente riconosciuto dal sistema.

U

Utente	Generico utilizzatore del sistema.
--------	------------------------------------

V

Visitatore	Utente sconosciuto al sistema.
------------	--------------------------------

Bibliography

Roger S. Pressman. Copyright © 1991 The MacGraw-Hill Companies, S.r.l.. McGraw-Hill. *Principi di Ingegneria del software.*

Ian Sommerville. *Software Engineering*. Eighth Edition. Addison-Wesley. 2007.

J. Arlow and I. Neustadt. *UML and the Unified Process*. Addison Wesley. 2002.

Martin Fowler. *UML Distilled Third Edition* . 2003.

and Neil Pitman. *UML 2.0 in a Nutshell* . O'Reilly. 2005.

Craig Larman. *Applying UML and Patterns*. 2. Prentice-Hall. 2001.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. 1995.

Scott Ambler. *Agile Modeling*. Wiley. 2002.

Scoot W. Ambler. *The Agile Unified Process (AUP)*. URL:<http://www.ambysoft.com/unifiedprocess/agileUP.html>. 2009.

Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley. 2000.

Kent Beck and Martin Fowler. *Planning Extreme Programming*. Addison-Wesley. 2000.

Deepak Alur, John Crupi, and Dan Malks. *Core J2EE Patterns*. Prentice-Hall. 2001.

Bill Burke and Richard Monson-Haefel. *Enterprise JavaBeans, 3.0*. O'Reilly. May 2006.

Brian Stansberry and Galder Zamarreno. *JBoss Application Server 5 Clustering Guide*. Nov 2008.

JBoss Community. *JBoss Application Server Installation And Getting Started Guide*. Feb 2009.

Chris Schlager and Marc Ruhrschnick. *The TaskJuggler Manual*.

Norman Walsh and Leonard Muellner. *DocBook*. The Definitive Guide. 2008. <http://www.docbook.org/tdg5/en/html/docbook.html> .