


## Appalto concorso sistema BR-jsys

Data: 6 novembre 2007	<b>Appalto concorso per la realizzazione di un sistema software per la gestione di <i>Business Rules</i> in ambito gestionale</b>
Autore: Gregorio Piccoli	
Azienda proponente: Zucchetti	
	
<b>BR-jsys</b>	

### Art.1) Oggetto dell' appalto

Il presente capitolato concerne l' affidamento a fornitori terzi, della realizzazione di un sistema software prototipale per la gestione ed esecuzione di “*business rules*” in ambito gestionale.

Il termine “**BR-jsys**” denota il prodotto in oggetto.

Il termine “committente” denota i proff. Vardanega e Conte, in qualità di delegati rappresentanti dell' azienda che ha commissionato il lavoro (di seguito denominata “proponente”).

### Art. 2) Caratteristiche e requisiti tecnici minimi

#### Premessa

Il progetto di seguito illustrato è intitolato:

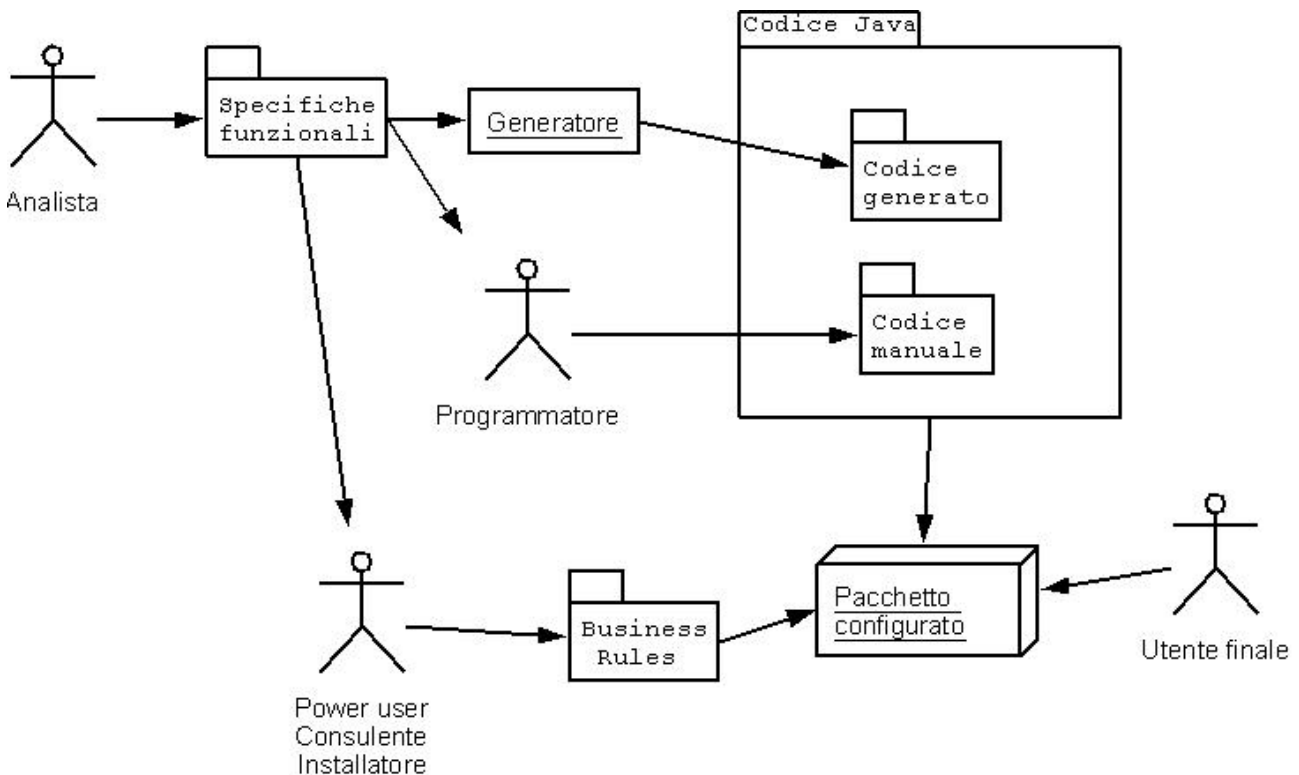
#### ***Business Rules per sistemi gestionali in architettura J2EE*** **BR-jsys**

Tale progetto, nel seguito denominato “progetto specifico”, si colloca nel quadro di un progetto aziendale strategico più ampio, nel seguito denominato “progetto globale”.

#### Obiettivi del progetto globale

Il progetto specifico fa parte di un più ampio progetto globale intrapreso dall' azienda committente, i cui obiettivi sono i seguenti:

- automazione nello sviluppo di sistemi informatici “gestionali” transazionali basati su *database* relazionali
- generazione automatica del codice a partire da modelli ottenuti nelle fasi di analisi
- graduale spostamento delle attività di realizzazione delle applicazioni dalla fase di codifica a quella di analisi
- realizzazione di “piattaforme applicative”, cioè procedure altamente configurabili ed estendibili a basso costo e con basse competenze tecnologiche da parte di utenti esperti o rivenditori ed installatori.



In questo scenario un punto di particolare importanza è la validazione dei dati prima della transazione verso il *database*. I sistemi relazionali effettuano le validazioni dei dati considerati staticamente, le procedure transazionali sono coinvolte nei processi applicativi. È quindi fondamentale poter configurare ed estendere la fase di validazione del dato che sta per essere inserito nella base dati.

Il progetto specifico si inserisce in questo contesto e si prefigge di realizzare un sistema di Regole di *Business* da associare a *Business Objects* (rappresentati da classi Java) a tempo d'esecuzione.

Una "Regola di *Business*" è un'espressione Booleana che verifica la validità dei dati contenuti in una classe e fornisce un eventuale messaggio di errore.

Possono essere create delle regole che non sono di validazione ma di calcolo, per completare gli eventuali *input* lasciati non specificati dall'utente.

Le classi Java che costituiscono i *Business Object* possono essere molto complesse, con vettori o tabelle di *hash* nel caso di oggetti di tipo Master/Detail.

Gli oggetti di tipo Master/Detail sono molto comuni nei programmi gestionali: fatture, ordini, listini etc. sono tutti organizzati in questo modo. Il sistema, parte del progetto globale, per questo tipo di oggetti produce delle classi che contengono dei vettori.

Nella modellazione dei dati per i database relazionali le entità vengono suddivise attraverso il processo di normalizzazione in più tabelle, dove una tabella padre guida le varie tabelle figlie; in questo caso troveremo una classe principale che modella la tabella principale e contiene altre classi che modellano le "weak entity" di tipo figlio.

Le classi dei *Business Object* devono essere esplorate dal sistema di *Business Rules* tramite "reflection"<sup>1</sup> raggiungendo ogni possibile elemento di questa struttura articolata.

Poiché si prevede che il sistema sia utilizzato da utenti poco esperti, una componente fondamentale del prodotto richiesto sarà un validatore che analizzando l'insieme delle regole verifichi se

<sup>1</sup> La *reflection* è la capacità di un programma di eseguire elaborazioni che hanno per oggetto il programma stesso, e in particolare la struttura del suo codice sorgente. Un programma Java in esecuzione, per esempio, può esaminare le classi da cui è costituito, i nomi e le *signature* dei loro metodi, e così via. (Si veda il *package* `java.lang.reflection`.)

costituiscono un insieme soddisfacibile o se rigetteranno qualunque insieme di valori possibile per i *Business Object*.

Il sistema dovrebbe fornire i seguenti benefici:

- elevata configurabilità ed estendibilità dei *business object*, mantenendo il rispetto dei controlli cablati.
- spostare l'attenzione della realizzazione del codice all'analisi dei requisiti
- aiutare gli utenti meno esperti a non commettere errori grossolani attraverso la verifica formale delle regole.

### Stato dell' arte

L'azienda committente inserirà **BR-jsys** nel proprio sistema di generazione automatica del codice a completamento delle fasi transazionali.

L'azienda è disposta a fornire delle classi di esempio di *Business Object* di varia complessità, dalla semplice gestione di tabelle secondarie a oggetti complessi come ordini o fatture al fine di calare **BR-jsys** in contesti di utilizzo reale.

L'azienda ha già effettuato delle analisi sul possibile linguaggio per esprimere le *Business Rules* ed è disponibile a fornire i risultati di questa fase.

E' inoltre disponibile a offrire un corso iniziale per l'avviamento del progetto della durata massima di due giornate lavorative.

### Obiettivi del progetto specifico

Nel contesto del progetto globale, il progetto specifico ha i seguenti obiettivi:

- realizzazione di un linguaggio per definire le regole di validazione e calcolo per classi J2EE
- realizzazione di un *repository* di regole (possibilmente in formato XML)
- realizzazione di un validatore che verifichi la coerenza delle regole
- realizzazione di API per eseguire le regole di validazione e ottenere il risultato della valutazione, sia come accettazione dei valori o meno che come messaggi di errore da presentare all'utente.
- realizzazione di un interprete che valuti le regole lette dal *repository* nel contesto della classe che realizza il *business object* e che indichi se è possibile proseguire con la transazione verso il database.

Il progetto ha anche un obiettivo secondario, di grande importanza per il prodotto finale, ma giudicato di difficile realizzazione nei vincoli di tempo e di costo del presente capitolato:

- realizzazione di un compilatore per tradurre le *business rules* in classi Java, per non avere penalizzazioni in tempi di esecuzione.

Le macro-fasi del progetto dovranno essere le seguenti:

- definizione del linguaggio di *Business Rules*
  - confronto con possibili standard di mercato
  - valutazione dell'espressività del linguaggio scelto, più possibile vicino ad esperienze di un utente evoluto (esempio: foglio di calcolo)
  - definizione di un *repository* di regole
- sviluppo di un prototipo, scomposto nelle tre componenti fondamentali:
  - *repository* in formato XML
  - validatore statico di coerenza

- interprete dinamico che associ classe Java e regole
- realizzazione del prodotto finale tramite:
  - sviluppo del prodotto
  - *test* su casi significativi, valutazione delle *performance* in termini di tempi di esecuzione
  - verifica, validazione e documentazione.

Per l'intera durata del progetto il fornitore potrà avere accesso tramite una linea di posta elettronica diretta a un referente designato dal "proponente", a disposizione per chiarimenti mentre sarà valutato nel merito, caso per caso, ogni altro tipo di intervento.

Sarà possibile prevedere riunioni tecniche di verifica dello stato di avanzamento dei lavori tra il fornitore e il "proponente", con cadenza almeno mensile, e per l'assunzione di decisioni fondamentali per il prosieguo del progetto (in particolare dopo la fase del prototipo iniziale).

### Art. 3) Documentazione

Il progetto **BR-jsys** vede nella documentazione uno dei punti fondamentali del proprio successo.

Dovranno essere documentati con particolare attenzione:

- il linguaggio di definizione delle *Business Rules*
- l'interazione con la classe Java
- il *repository* delle regole
- le API di attivazione delle regole
- l'interprete di esecuzione delle regole
- il validatore di coerenza delle regole.

Con la consegna di **BR-jsys** il fornitore assumerà anche l'onere di fornire i manuali e ogni altra documentazione tecnica idonea ad assicurare il soddisfacente funzionamento e manutenzione del prodotto, compreso il codice sorgente adeguatamente commentato.

### Art. 4) Rinvio

Per tutto quanto non previsto nel presente capitolato, sono applicabili le vigenti disposizioni di legge.

### Riferimenti

Per avere un'idea di cosa si può intendere per *Business Rules* possono essere utili questi riferimenti a informazioni Web o prodotti commerciali:

<http://www.businessrulesgroup.org/brmanifesto.htm>

<http://www.brcommunity.com/europe.php>

[http://en.wikipedia.org/wiki/Business\\_rules](http://en.wikipedia.org/wiki/Business_rules)

<http://www.ilog.com/products/businessrules/>