

Università degli Studi di Roma “La Sapienza”
Facoltà di Ingegneria – Corso di Laurea in Ingegneria Gestionale
Corso di Progettazione del Software
Proff. Toni Mancini e Monica Scannapieco

Progetto **PC.20070906**

versione del 20 settembre 2007

Si vuole progettare e realizzare *CostChecker*, un sistema informatico, da usarsi in una Università, per la gestione dei rendiconti di progetti di ricerca. Il sistema deve permettere la memorizzazione dei progetti e delle spese effettuate sui relativi fondi. Su richiesta, il sistema deve consentire di rendicontare tutte le spese effettuate per un dato progetto in un dato periodo temporale.

Si richiede di effettuare la fase di Analisi, producendo uno schema concettuale per l'applicazione i cui requisiti sono descritti in calce.

Requisiti

Il sistema *CostChecker* deve tenere traccia dei diversi progetti finanziati nell'Università. Di ogni progetto interessa il codice, il nome, la data di inizio e quella di fine, oltre che il personale che vi partecipa. I partecipanti ad un progetto possono essere sia dipendenti strutturati dell'Università che personale non strutturato. Di ogni partecipante (strutturato o no) è necessario conoscere nome e cognome, *costo giornaliero* (ovvero quanto costa una sua giornata di lavoro) e impegno complessivo (in termini di giorni/uomo) per i singoli progetti a cui partecipa.

Oltre che per pagare il personale, i fondi assegnati ad un progetto vengono utilizzati per effettuare delle spese, che sono identificate mediante un codice e possono essere di tre tipologie: spese di acquisto di beni materiali, spese per contratti di consulenza, e spese per viaggi (*missioni*) che possono essere effettuati –esclusivamente dai suoi partecipanti– per partecipare a convegni.

In particolare, i contratti di consulenza permettono di affidare particolari attività relative al progetto a professionisti esterni (consulenti). Di un contratto di consulenza interessa conoscere, oltre che relative date di inizio e fine, il professionista incaricato. Si noti che, per ragioni normative, un consulente non può essere dipendente (strutturato o no) dell'Università. Anche dei consulenti esterni interessa conoscere nome, cognome e costo giornaliero, da cui viene calcolato l'importo della consulenza (come prodotto tra costo giornaliero e durata del contratto).

Per quanto concerne invece le spese per acquisto di beni materiali, di queste va mantenuta informazione circa prezzo e data di acquisto. Infine, per quanto riguarda le spese per missioni, oltre all'importo e alla data in cui il partecipante ha ottenuto il rimborso, interessa conoscere titolo, sede e date di inizio e fine del convegno in questione.

Il sistema deve permettere di effettuare la *rendicontazione* di un progetto in un dato periodo di tempo (detto *periodo di rendicontazione*) come somma degli importi di tutte le spese (relative al pagamento di personale universitario o di consulenti esterni, di spese per acquisto di beni e per missioni) di competenza del periodo.

La rendicontazione delle varie voci, nel periodo in esame, avviene come segue:

Consulenze esterne Il costo ai fini della rendicontazione di una consulenza è dato dal costo giornaliero del titolare del contratto per il numero di giorni del contratto che ricadono nel periodo di rendicontazione.

Spese di acquisto di beni materiali Tali spese contribuiscono (con il prezzo d'acquisto) al rendiconto solo nel caso in cui il periodo di rendicontazione comprenda la data di acquisto.

Personale universitario Il costo ai fini della rendicontazione del personale dell'Università è più complesso da calcolarsi, dato che è diverso per dipendenti strutturati e non strutturati.

In particolare, per un dipendente strutturato, il costo *giornaliero* ai fini della rendicontazione è pari al suo costo giornaliero maggiorato del 60%, mentre per i non strutturati tale maggiorazione è solo del 20% (tale incremento è dovuto a fattori fiscali e previdenziali).

Il costo totale ai fini della rendicontazione per il periodo richiesto dipende da tale costo giornaliero, dalla durata complessiva del progetto e dall'impegno complessivo del lavoratore nel progetto. In particolare, viene assunto che l'impegno sia *uniforme* durante tutta la durata del progetto: quindi, ad esempio, il costo rendicontato per un periodo di 30 gg di un dipendente che ha dichiarato un impegno complessivo di 120 gg in un progetto di durata totale pari a 360 gg è dato da $120/360 \cdot 30 = 10$ volte il suo costo giornaliero ai fini della rendicontazione (a sua volta pari al suo costo giornaliero maggiorato del 20% o del 60%).

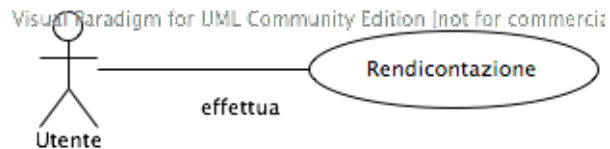
Missioni per partecipazione a convegni Le missioni contribuiscono (con l'importo rimborsato al partecipante) al rendiconto solo nel caso in cui la data di rimborso delle spese cada nel periodo di rendicontazione. Tuttavia, affinché una missione possa essere rendicontata su un progetto è necessario che il convegno abbia affinità con i suoi obiettivi. Per consentire tali confronti, sia gli obiettivi di un progetto che i temi di un convegno

vengono rappresentati da un insieme di keyword (stringhe). Il sistema deve dichiarare un convegno *affine* ad un progetto se e solo se almeno il 70% delle keyword che ne caratterizzano gli obiettivi sono comprese tra i temi del convegno stesso.

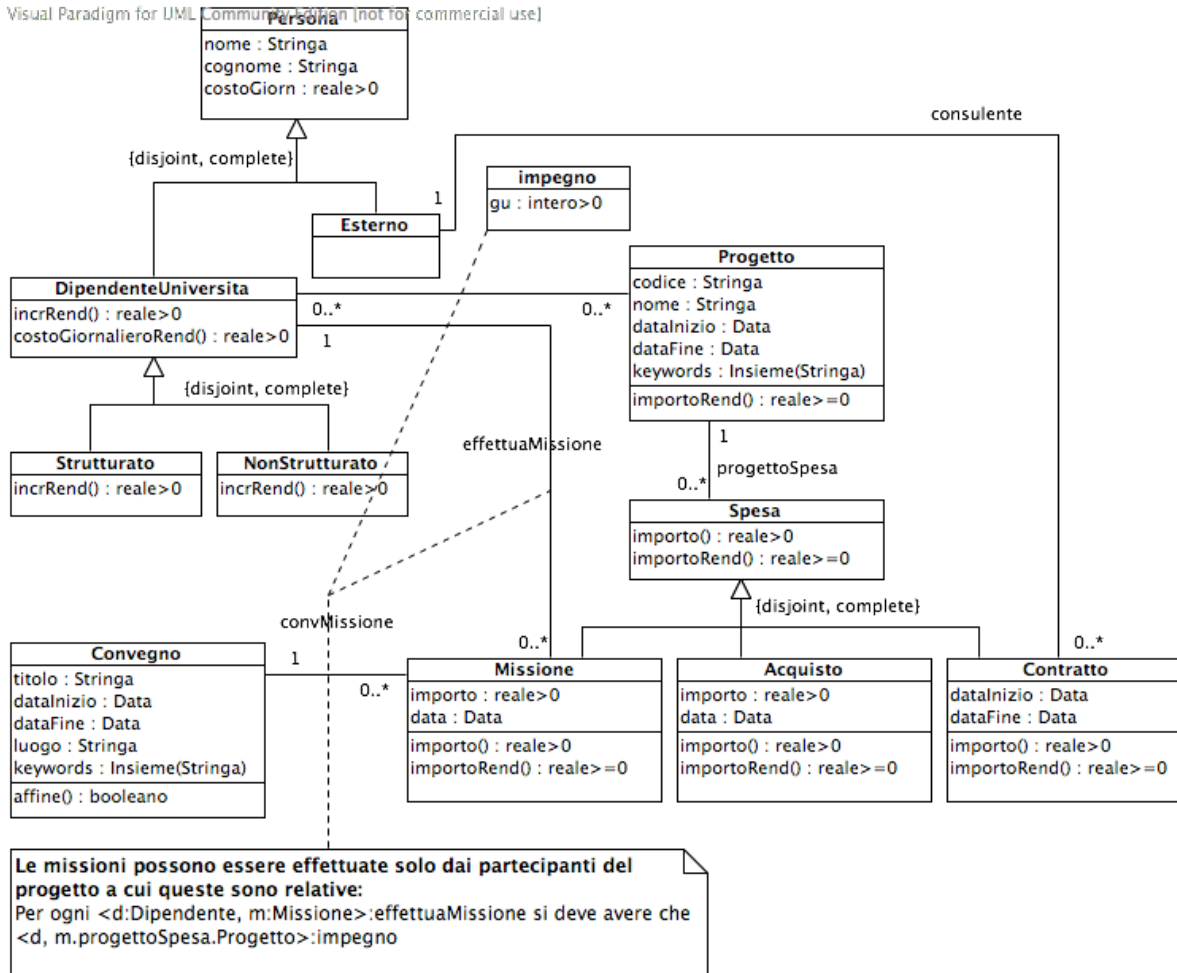
Se il convegno non è affine al progetto, il costo della missione non può venire rimborsato al dipendente.

1 Fase di Analisi

1.1 Diagramma degli Use Case



1.2 Diagramma delle classi UML



1.3 Specifica dei tipi di dato

Nessun tipo di dato definito

1.4 Specifica degli use case

SpecificaUseCase Rendicontazione

```

rendiconto(p:Progetto, dataInizio:Data, dataFine:Data): reale>=0
pre: dataInizio <= dataFine
    
```

```
post: result = p.importoRend(dataInizio, dataFine)
```

FineSpecifica

1.5 Specifica delle classi e diagrammi degli stati e transizioni

La classe DipendenteUniversita e le sue sottoclassi

SpecificaClasse DipendenteUniversita

```
costoGiornalieroRend(): reale > 0
```

```
pre: nessuna
```

```
post: result = this.costoGiornaliero * this.incrRend()
```

```
incrRend(): reale > 0
```

```
pre: nessuna
```

```
post: result dipende dalla classe piu' specifica a cui this appartiene
```

FineSpecifica

SpecificaClasse Strutturato

```
incrRend(): reale > 0
```

```
pre: nessuna
```

```
post: result e' pari a 1.6
```

FineSpecifica

SpecificaClasse NonStrutturato

```
incrRend(): reale > 0
```

```
pre: nessuna
```

```
post: result e' pari a 1.2
```

FineSpecifica

La classe Progetto

SpecificaClasse Progetto

```
importoRend(dataInizio:Data, dataFine:Data):reale>=0
```

```
pre: dataInizio <= dataFine
```

```
post:
```

```
Sia gg = min(dataFine, this.dataFine).differenza(  
max(dataInizio, this.dataInizio), GIORNI).
```

result e' pari a:

$$\sum_{l \in \text{this.progettoSpesa.l.Spesa}} \text{importoRend}() + \sum_{i \in \text{this.impegno}} [(i.gu \times \max(0, gg) / \text{this.dataFine.differenza}(\text{this.dataInizio}, \text{GIORNI})) \times i.DipendenteUniversitario.\text{costoGiornalieroRend}()]$$

FineSpecifica

La classe Spesa e le sue sottoclassi

Specificazione Classe Spesa

importo(): reale > 0

pre: nessuna

post: result dipende dalla sottoclasse piu' specifica a cui this appartiene

importoRend(dataInizio:Data, dataFine:Data): reale >= 0

pre: dataInizio <= dataFine

post: result dipende dalla sottoclasse piu' specifica a cui this appartiene

FineSpecifica

Specificazione Classe Missione

importo(): reale > 0

pre: nessuna

post: result = this.importo

importoRend(dataInizio:Data, dataFine:Data): reale >= 0

pre: dataInizio <= dataFine

post: se this.data >= dataInizio e this.data <= dataFine e

 this.convMissione.Convegno.affine(this.progettoSpesa.Progetto)

 allora result = this.importo()

 altrimenti result = 0.

FineSpecifica

Specificazione Classe Acquisto

importo(): reale > 0

pre: nessuna

post: result = this.importo

importoRend(dataInizio:Data, dataFine:Data): reale >= 0

```
pre: dataInizio <= dataFine
post: se this.data >= dataInizio e this.data <= dataFine
      allora result = this.importo()
      altrimenti result = 0.
```

FineSpecifica

SpecificaClasse Contratto

```
importo(): reale > 0
pre: nessuna
post: result = this.consulente.Esterno.costoGiornaliero *
      this.dataFine.differenza(GIORNI, this.dataInizio).
```

```
importoRend(dataInizio:Data, dataFine:Data): reale >= 0
pre: dataInizio <= dataFine
post:
  sia gg = min(dataFine, this.dataFine).differenza(
           max(dataInizio, this.dataInizio), GIORNI).
  result = this.importo() * max(0,gg) /
           this.dataFine.differenza(GIORNI, this.dataInizio).
```

FineSpecifica

La classe Convegno

SpecificaClasse Convegno

```
affine(p:Progetto):booleano
pre: nessuna
post:
  Sia kwcomuni = this.keywords intersezione p.keywords.
  Se |kwcomuni| >= 0.7 * |p.keywords|, result = true, altrimenti result = false.
```

FineSpecifica