

Disegno del sistema

Release 0.5

December 8, 2009

Firenze



Approvazione, redazione, lista distribuzione

approvato da	il giorno	firma
Marco Tinacci	December 8, 2009	

redatto da	il giorno	firma
Francesco Calabri	December 8, 2009	
Niccolò Rogai	December 8, 2009	
Marco Tinacci	December 8, 2009	

distribuito a	il giorno	firma
Francesco Calabri	December 8, 2009	
Manuele Paulantonio	December 8, 2009	
Daniele Poggi	December 8, 2009	
Massimo Nocentini	December 8, 2009	
Niccolò Rogai	December 8, 2009	
Marco Tinacci	December 8, 2009	

Contents

0.1	Introduzione	3
0.2	Descrizione architetturale	4
1	Class Diagrams	6
1.1	Commons	6
1.2	Chart Generator	9
1.3	Task Data	10
1.4	Gif Area	11
1.5	User Options	12
1.6	Reports	13

0.1 Introduzione

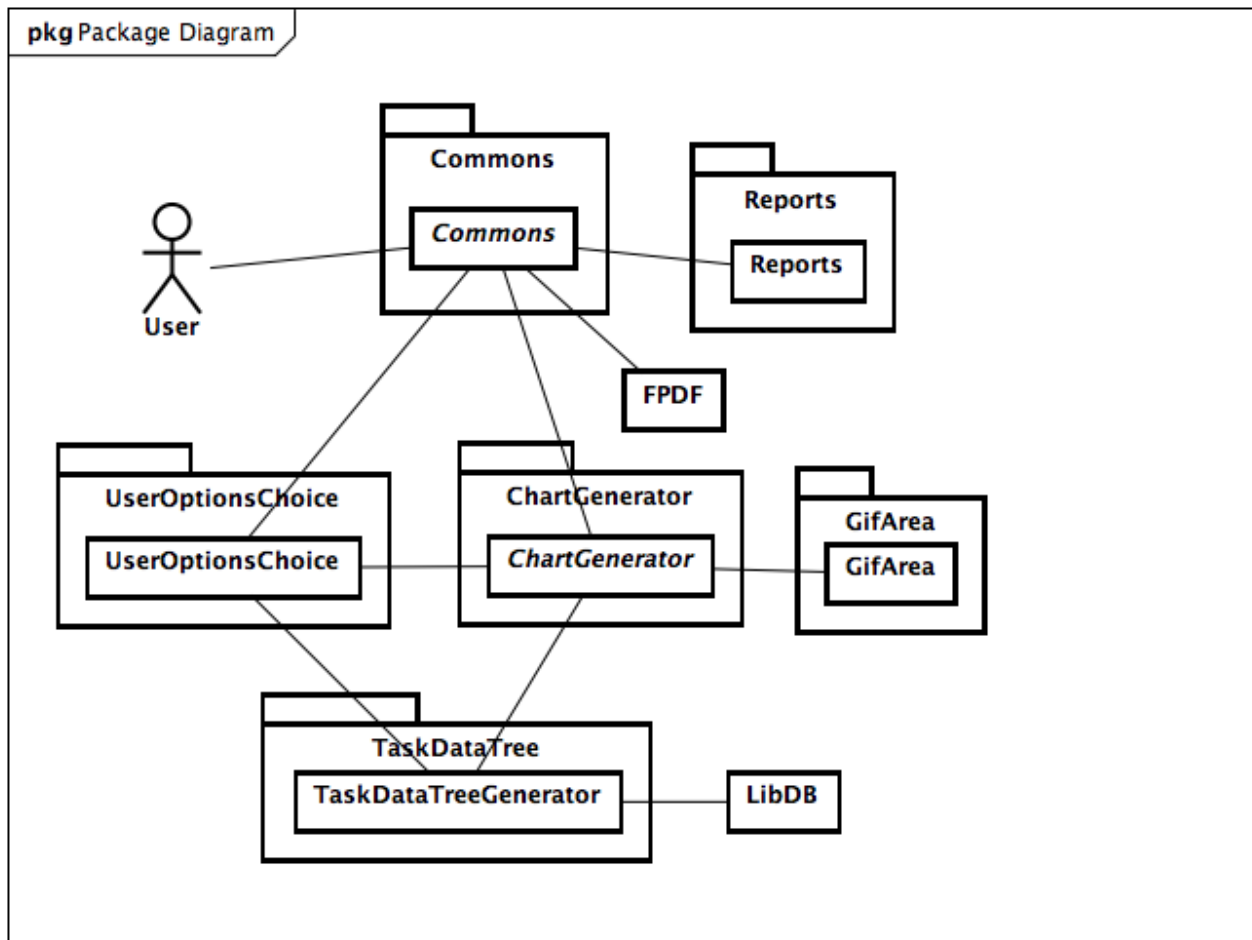
In questo documento viene illustrata l'architettura del modulo tramite le viste dei *class diagrams* e i *sequence diagrams*, come evoluzione naturale, rispettivamente di *domain model* e *use cases*.

I diagrammi delle classi servono quindi a individuare le strutture dei dati e i metodi associati mentre quelli di sequenza evidenziano le comunicazioni che intercorrono tra le entità.

Per semplificare la rappresentazione del diagramma delle classi abbiamo deciso di scrivere gli attributi privati con la notazione di quelli pubblici “+ attribute” dando per assunto che siano provvisti dei metodi pubblici di “getAttribute()” e “setAttribute()”.

Quando si fa riferimento a un file, a meno che non sia espressamente specificato, è sempre un path che inizia implicitamente dalla root path dove PMango è installato.

0.2 Descrizione architetturale



Il diagramma propone una visuale ad alto livello delle componenti principali del modulo.

- Commons: questa componente è quella che controlla il flusso principale e si interfaccia direttamente con l'utente (la pressione dei pulsanti dell'interfaccia utente corrisponde a una chiamata di un metodo). Utilizza la componente ChartGenerator per generare i diagrammi richiesti e UserOptionsChoice per gestire la visualizzazione e l'acquisizione delle opzioni utente presenti nella form della pagina. Per includere il diagramma in un PDF è necessario che comunichi con la libreria FPDF, mentre per aggiungere i diagrammi al report deve utilizzare la componente Reports.
- ChartGenerator: questa componente si occupa della generazione dei diagrammi. L'elaborazione grafica viene delegata alla componente GifArea, l'acquisizione dei dati a TaskDataTreeGenerator, e le opzioni utente vengono lette da UserOptionsChoice.
- TaskDataTreeGenerator: questa componente si occupa della generazione di una strut-

tura dati contenente tutte le informazioni dei task di cui è composto il progetto. Si interfaccia con il database per il reperimento dei dati utilizzando la classe già presente in PMango *DBQuery*. Le richieste di quali dati verranno richiesti al database dipendono dalle opzioni specificate in *UserOptionsChoice*.

- *UserOptionsChoices*: questa componente gestisce i campi di un array associativo che contiene le opzioni scelte dall'utente.
- *GifArea*: componente che contiene utilità per l'elaborazione di immagini gif. Queste funzionalità si collocano in un livello medio tra la generazione del diagramma completo e il disegno di segmenti e label.
- *Reports*: questo componente si occupa di aggiungere il diagramma prodotto nel report.

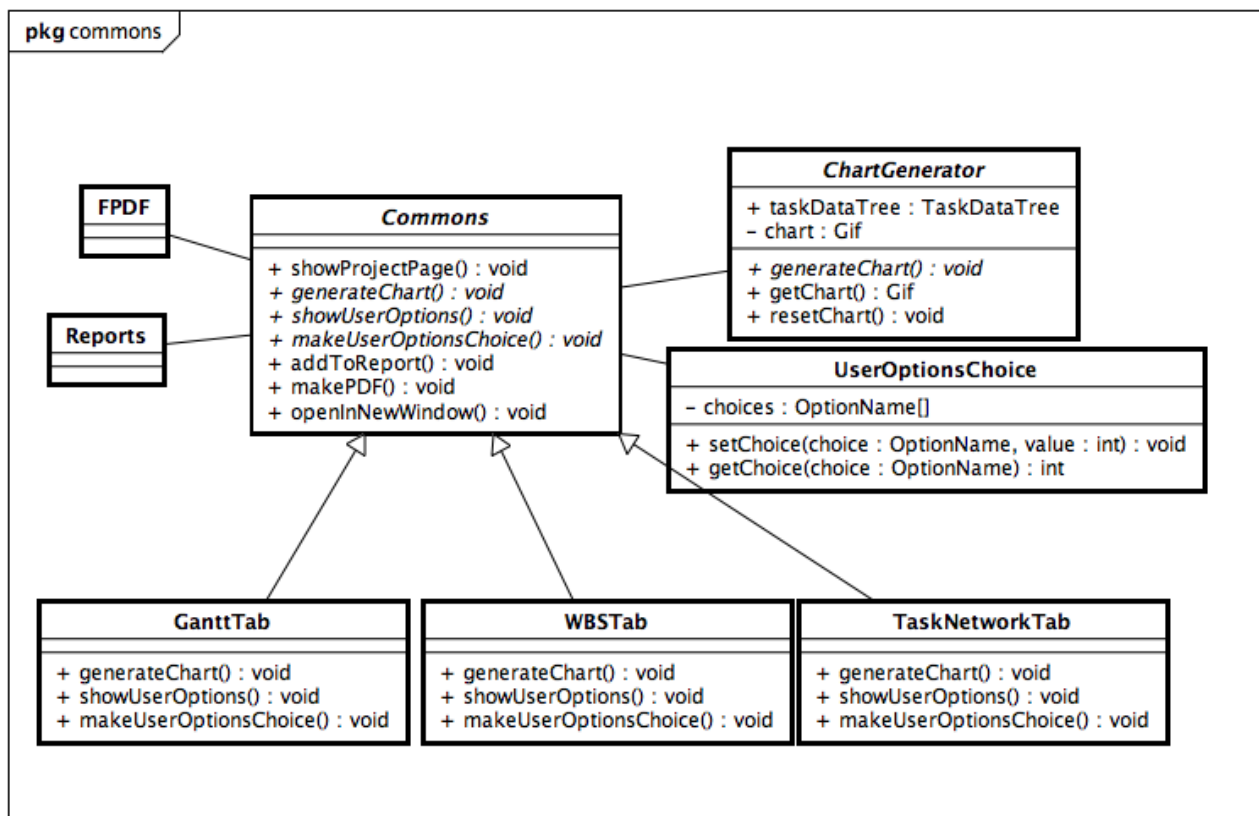
Le dipendenze esterne al modulo sono:

- FPDF: la libreria per la generazione dei PDF "lib/fpdf/fpdf.php"
- DBQuery: la classe per reperire le informazioni dal database "classes/query.class.php"

Chapter 1

Class Diagrams

1.1 Commons



La classe Commons racchiude tutte le funzioni principali del modulo. Utilizza le classi

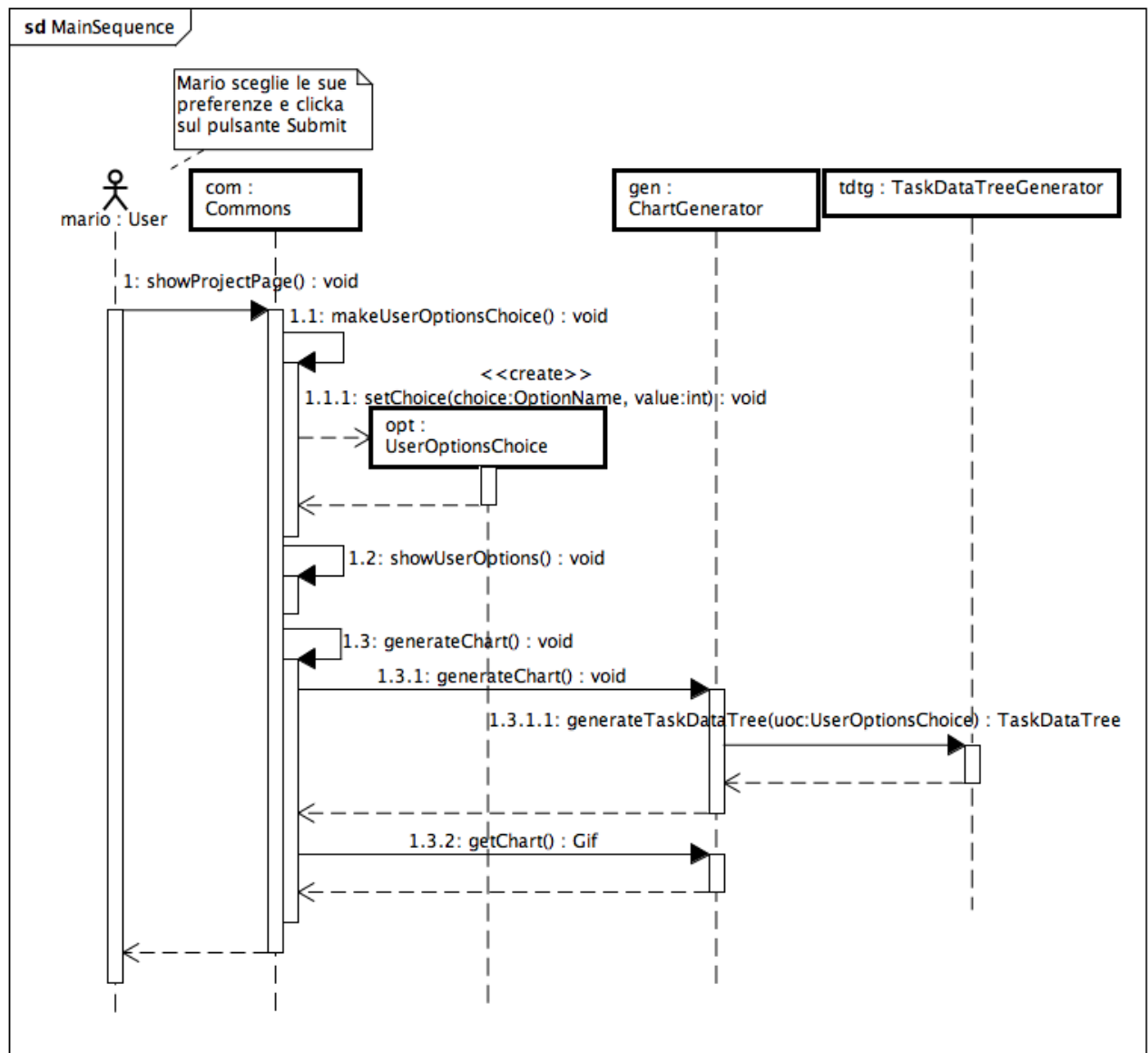
- ChartGenerator

- UserOptionsChoice
- Reports
- FPDF (esterna al modulo e già utilizzata in PMango)

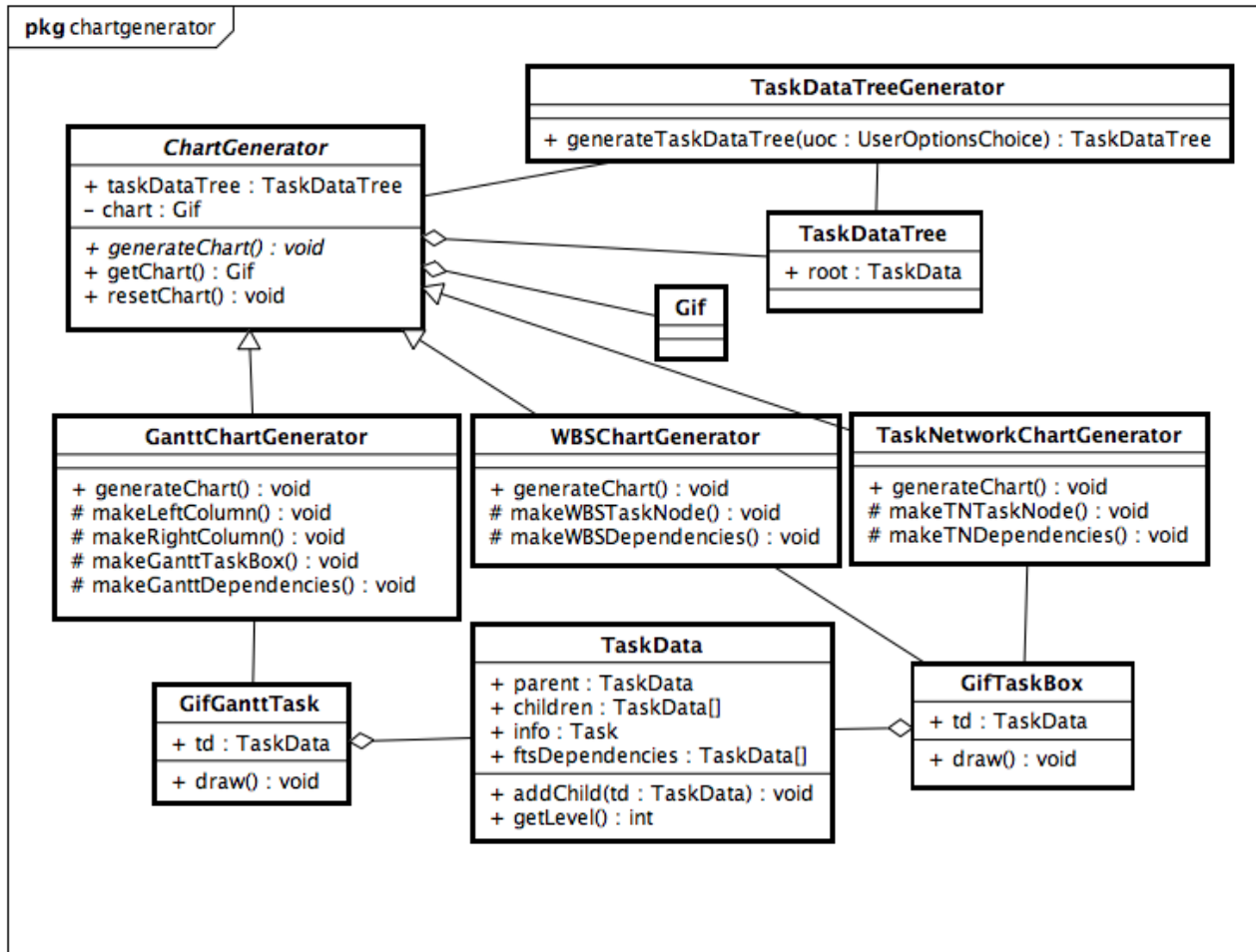
La classe è astratta a causa dei metodi astratti *generateChart*, *ShowUserOptions* e *makeUserOptionsChoice* (che dovranno essere implementati nelle successive estensioni). In generale i metodi contenuti nella classe Commons sono quelli presentati nel documento dei casi d'uso, contenente una descrizione più approfondita dei A seconda di quale tab viene caricato, può essere usata l'estensione di Commons corrispondente:

- GanttTab
- WBSTab
- TaskNetworkTab

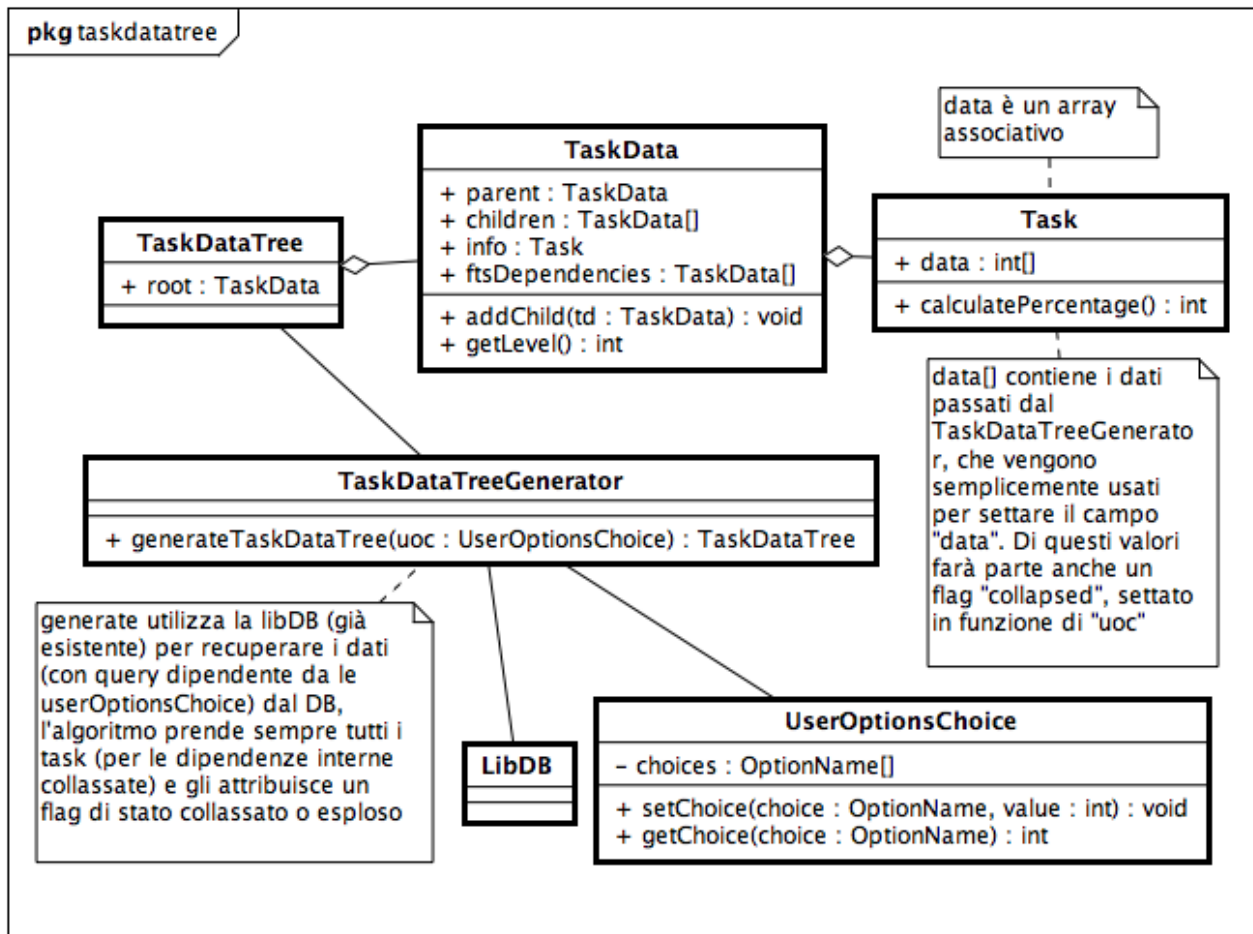
che implementano la generazione del diagramma, la creazione della maschera delle user options e l'acquisizione delle scelte dei parametri.



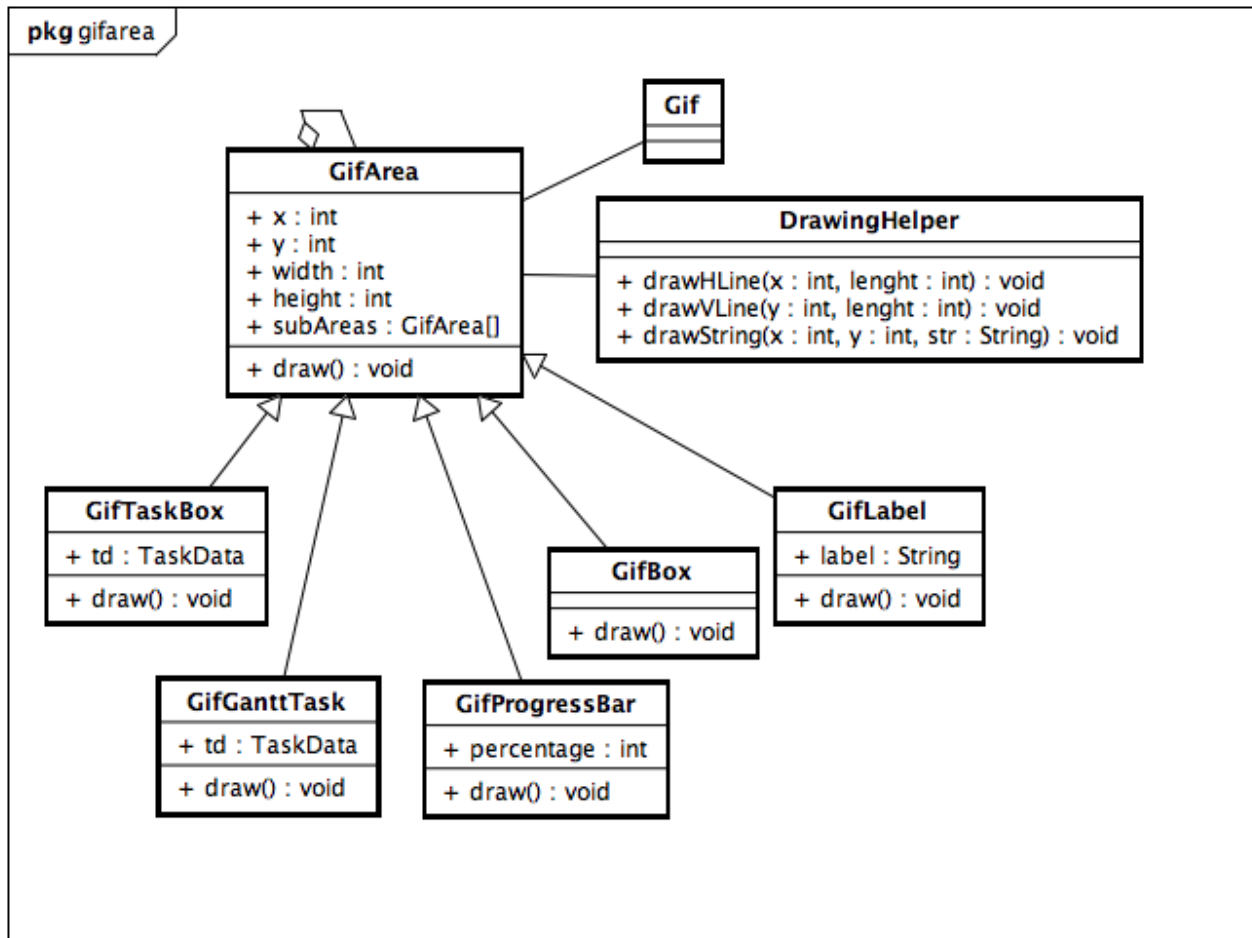
1.2 Chart Generator



1.3 Task Data

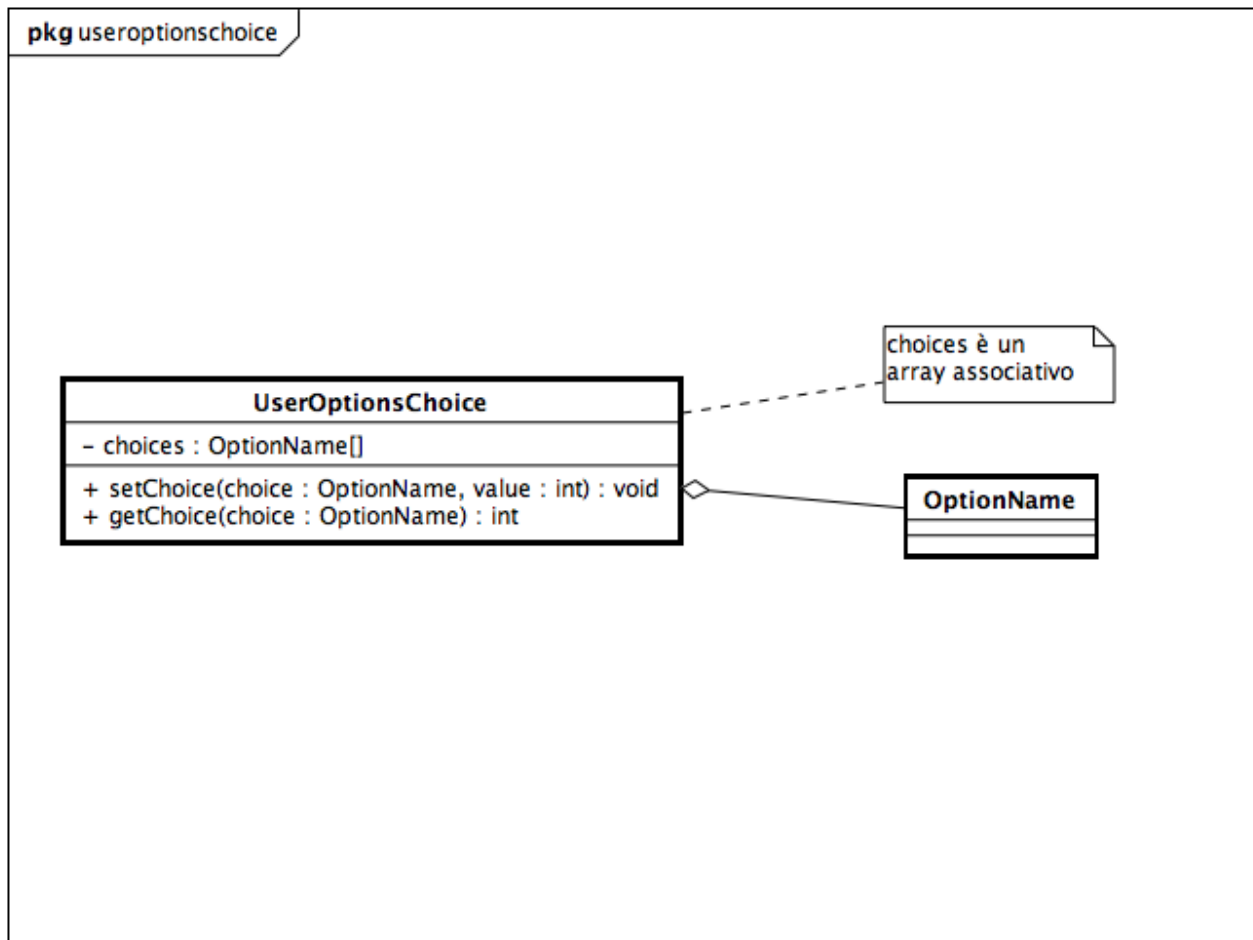


1.4 Gif Area



La **GifArea** e i suoi derivati realizzano il pattern *composite* che si presta a portare ad un livello più alto la gestione della grafica, infatti qualunque cosa che sia una “**GifArea**” avrà la possibilità di essere stampata (tramite la funzione *draw*) su una gif in un determinato punto (x, y) occupando una certa area di dimensioni “width” e “height”. I componenti più base per maneggiare la grafica in questo contesto sono la “**GifLabel**” che è la rappresentazione grafica di un testo, la “**GifBox**” che è una rappresentazione grafica di un rettangolo, la “**GifProgressBar**” che rappresenta una barra di completamento. Anche i **GifTaskBox** sono una **GifArea** e sono composti da dei **GifBox** con al suo interno delle **GifLabel**, e da una **GifProgressBar** quando richiesta, e il *draw* del **GifTaskBox** farà il *draw* di tutti questi suoi componentini interni. Una volta costruito l’albero dei taskdata sarà molto facile disegnare un **GifTask** da un taskdata. Così si separano la visualizzazione dei taskbox dal problema del posizionamento e l’unione con le linee.

1.5 User Options



- **UserOptionsChoice**: la classe che racchiude le scelte fatte dall'utente in un array associativo controllato dai metodi get e set.
- **OptionName**: un'enumerata che vincola l'accesso all'array associativo ai soli campi specificati. Questa contiene le seguenti voci (associate ai possibili valori che possono assumere):
 - **TaskNameOption**: boolean
 - **PlannedTimeFrameOption**: boolean
 - **ActualTimeFrameOption**: boolean
 - **PlannedDataOption**: boolean
 - **ActualDataOption**: boolean

- CompletionBarOption: boolean
- AlertMarkUserOption: boolean
- WBSExplosionLevelUserOption: numeric
- WBSUserSpecificationUserLevel: boolean
- ReplicateArrowUserOption: boolean
- FindCriticalPathUserOption: boolean
- ResourcesDetailsOption: boolean
- CompleteDiagramUserOptions: boolean
- WBSTreeSpecification: LevelSpecification, UserCustomSpecification
- TimeGrain: HourlyGrain, DailyGrain, WeaklyGrain, MonthlyGrain, AnnuallyGrain
- ImageDimensionUserOption: CustomDim, FitInWindowDim, OptionalDim, DefaultDim
- TimeRangeUserOption: CustomRange, WholeProjectRange, FromStartRange, ToEndRange

La semantica di questi parametri è specificata nel domain model. Tutti i parametri boolean sono intesi *false* se non presenti, è quindi necessario un doppio controllo per capire se è impostato il valore *true*: il controllo sulla presenza del campo ed eventualmente sul suo valore.

1.6 Reports