

Use cases

Release 1.0

November 16, 2009

Firenze



Approvazione, redazione, lista distribuzione

approvato da	il giorno	firma
Marco Tinacci	16/09/2009	

redatto da	il giorno	firma
Francesco Calabri	16/09/2009	
Manuele Paulantonio	16/09/2009	
Massimo Nocentini	16/09/2009	

distribuito a	il giorno	firma
Daniele Poggi	16/09/2009	
Niccoló Rogai	16/09/2009	
Marco Tinacci	16/09/2009	

Contents

1	Commons	6
1.1	Make Dependencies	6
1.1.1	Basic course	6
1.1.2	Alternative course	6
1.2	Make NodeTaskbox	6
1.2.1	Basic course	6
1.2.2	Alternative course	8
1.3	Show Project Page	8
1.3.1	Basic course	8
1.4	Generate Chart	9
1.5	Show UserOptions	9
1.6	Make UserOptionsChoice	9
1.6.1	Basic course	9
1.7	User Action	9
1.8	Add to Report UserAction	9
1.8.1	Basic course	9
1.9	Make PDF	10
1.9.1	Basic course	10
1.10	Refresh Chart	10
1.10.1	Basic course	10
1.11	Select User Option	10
1.11.1	Basic course	10
1.12	Open in New Window	11
1.12.1	Basic course	11
2	Gantt chart	12
2.1	Make Left Column	12
2.1.1	Basic course	12
2.1.2	Alternative course	12
2.2	Make GanttTaskbox	13
2.2.1	Basic course	13

2.2.2	Alternative course	14
2.3	Make Right Column	14
2.3.1	Basic course	14
2.3.2	Alternative course	14
2.4	Generate Gantt Chart	15
2.4.1	Basic course	15
3	WBS chart	16
3.1	Make Hierarchycal Dependencies	17
3.1.1	Basic course	17
3.1.2	Alternative course	17
3.2	Generate WBS Chart	17
3.2.1	Basic course	17
4	TaskNetwork chart	18
4.1	Generate TaskNetwork Chart	19
4.1.1	Basic course	19
4.2	Create Critical Path Table	19
4.2.1	Basic course	19

Introduzione

Descrizione dell'acronimo: **cdns** sta per "come **d**escritto **n**ella **s**pecifica".

Entire System UML diagram

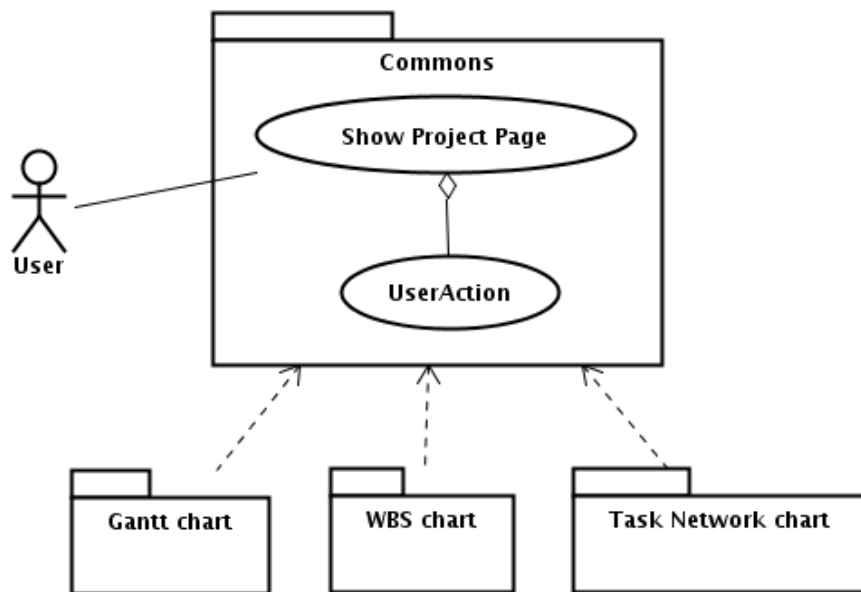


Figure 1: Entire system UML diagram

Chapter 1

Commons

1.1 Make Dependencies

1.1.1 Basic course

Si assume che l'insieme di *TaskBox* sia già costruito.

Il client richiede di rappresentare graficamente le *Finish-ToStartDependency* relative all'insieme di *TaskBox* già costruite.

Il sistema esegue questi passi:

1. costruisce un insieme *Dep* di coppie del tipo (a, b) , con $a, b \in TaskBox$, tale che $(a, b) \in Dep \Leftrightarrow b$ non può iniziare finchè a non è stata completata.
2. per ogni coppia $(a, b) \in Dep$ costruisci una linea spezzata cdns.

1.1.2 Alternative course

not well formed project Se la struttura albero WBS del progetto non è ben formata allora si deve cercare di dare la migliore euristica possibile per la rappresentazione delle dipendenze.

1.2 Make NodeTaskbox

1.2.1 Basic course

Il client prende un reference al *WBSChartGenerator* e domanda di creare la rappresentazione grafica di un *Task*.

Il sistema esegue questi passi:

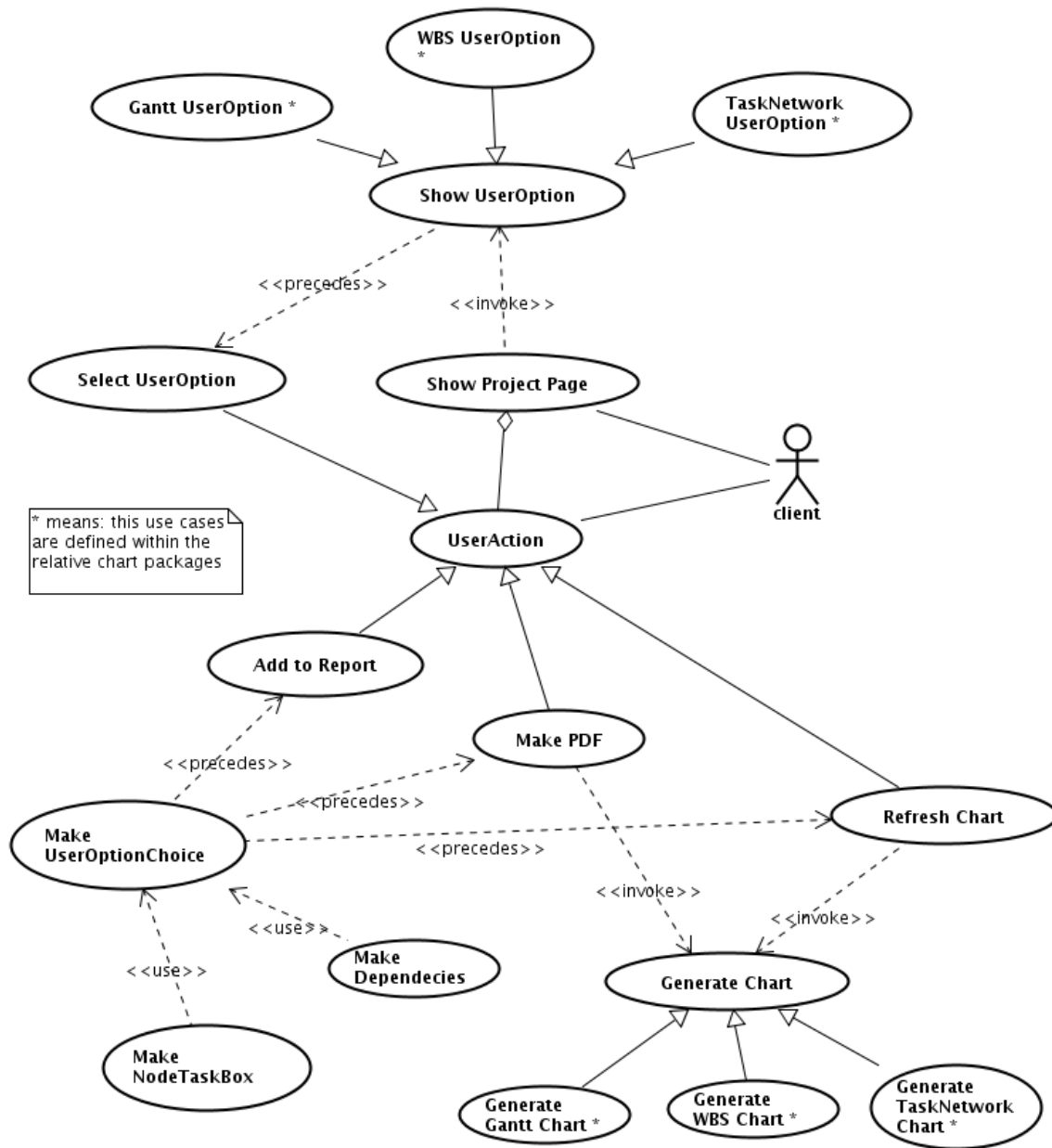


Figure 1.1: Overall Commons UML diagram

1. recupera il *Task* da rappresentare.
2. costruisce la rappresentazione grafica, il *NodeTaskbox* in base alla scelta *WBSTreeSpecification*. Il *NodeTaskbox* è una composizione di *Strip*. Il sistema costruisce una biezione tra *BoxedStrip* e le scelte presenti in *UserOptionsChoice* in questo modo:

- se *UserOptionsChoice* contiene *TaskNameOption*, allora costruisce una *Strip* contenente il nome del *Task* cdns.
- se *UserOptionsChoice* contiene *ResourcesDetailsOption*, allora sul margine destro della *GanttTaskBox* appendi la stringa contenente la lista delle risorse cdns. Altrimenti appendi sul margine destro l'effort cdns.
- se *UserOptionsChoice* contiene *PlannedTimeFrameOption* allora costruisce due *Strip* adiacenti contenenti rispettivamente le date di inizio e fine *Task* cdns.
- se *UserOptionsChoice* contiene *ActualTimeFrameOption* allora costruisce due *Strip* adiacenti contenenti rispettivamente le date di inizio e fine *Task* reali cdns.
- se *UserOptionsChoice* contiene *PlannedDataOption* allora costruisce tre *Strip* adiacenti contenenti rispettivamente la durata pianificata, lo sforzo complessivo pianificato, il costo pianificato del *Task* cdns.
- se *UserOptionsChoice* contiene *ActualDataOption* allora costruisce tre *Strip* adiacenti contenenti rispettivamente la durata "dall'inizio ad oggi", lo sforzo complessivo "ad oggi" effettuato, il costo complessivo "ad oggi" del *Task* cdns.
- se *UserOptionsChoice* contiene *CompletionBarOption* allora costruisce la barra di completamento del *Task* cdns.
- to complete with the missing options

1.2.2 Alternative course

troncamento del nome del *Task* se la stringa scritta supera la dimensione fissata nel documento di specifica, allora il sistema la tronca cdns.

1.3 Show Project Page

1.3.1 Basic course

Il client digita l'URL della *ProjectPage*¹.

Il sistema invia in risposta la pagina richiesta aggiungendo all'insieme dei tab presenti, tre tab relativi ai *Chart* descritti nel documento **Domain Model** e oggetto dell'appalto.

Il client fa click sul tab relativo al *Chart* che vuole generare.

Il sistema invoca la specializzazione dello usecase 1.5 relativa al *Chart* scelto, per costruire l'insieme delle possibili *UserOption*: dopo di che aggiunge questo insieme alla pagina di risposta.

Il client effettua una azione, invocando una specializzazione di 1.7.

¹TODO:definire *ProjectPage* nel documento dei Mockup

1.4 Generate Chart

Questo use case, come rappresentato nel diagramma UML Figure 1.1, fornisce il punto di astrazione per altri use case. Per questo motivo la descrizione del comportamento che si vuole modellare viene descritta in ogni specializzazione.

1.5 Show UserOptions

Questo use case, come rappresentato nel diagramma UML Figure 1.1, fornisce il punto di astrazione per altri use case. Usiamo questo formalismo per permettere ad ogni specializzazione di esprimere solo quali sono le *UserOption* disponibili per il relativo *Chart*.

1.6 Make UserOptionsChoice

1.6.1 Basic course

Il sistema riceve una http request contenente una sequenza di *UserOption* che sono state selezionate dall'utente. Costruisce una *UserOptionsChoice* così: per ogni *UserOption* indicata nella request si aggiunge all'oggetto in costruzione.

Lato server abbiamo l'insieme di scelte disponibile per le azioni successive.

1.7 User Action

Questo use case, come rappresentato nel diagramma UML Figure 1.1, fornisce il punto di astrazione per altri use case. Per questo motivo la descrizione del comportamento che si vuole modellare viene descritta in ogni specializzazione.

1.8 Add to Report UserAction

1.8.1 Basic course

Si assume che il client stia interagendo con il tab relativo al *Chart* che vuole generare.

Il client fa click sul pulsante "Add to Report".

Il sistema invoca lo use case 1.6 per costruirsi la *UserOptionsChoice*.

Il sistema aggiunge una *ReportSection* per aggiungere alla reportistica il *Chart* richiesto, con la relativa *UserOptionsChoice*. Queste sezioni saranno elencate nella schermata della reportistica già esistente.

1.9 Make PDF

1.9.1 Basic course

Si assume che il client stia interagendo con il tab relativo al *Chart* che vuole generare.

Il client fa click sul pulsante “Make PDF”.

Il sistema esegue questi passi:

- invoca lo use case 1.6 per costruirsi la *UserOptionsChoice*.
- esegue una ricerca dei *Task* che devono essere inclusi nel *Chart*.
- invoca la specializzazione di 1.4 per la creazione del relativo *Chart*
- costruisce un file pdf aggiungendo al suo interno la rappresentazione generata
- invia al client una pagina di risposta con una **icona** accanto ai due pulsanti della reportistica, per segnalare che il file PDF è disponibile.

1.10 Refresh Chart

1.10.1 Basic course

Si assume che il client stia interagendo con il tab relativo al *Chart* che vuole generare.

Il client fa click sul pulsante “Refresh”.

Il sistema esegue questi passi:

- invoca lo use case 1.6 per costruirsi la *UserOptionsChoice*.
- esegue una ricerca dei *Task* che devono essere inclusi nel *Chart*.
- invoca la specializzazione di 1.4 per la creazione del relativo *Chart*
- invia al client una pagina di risposta inserendo la rappresentazione nel bottom della pagina.

1.11 Select User Option

1.11.1 Basic course

Si assume che il client stia interagendo con il tab relativo al *Chart* che vuole generare, quindi lo use case 1.3 è già stato completato.

Il client vuole selezionare alcune *UserOption* per guidare la rappresentazione delle informazioni che verranno codificate nel *Chart*.

Queste sono rappresentate tramite una form html, sotto forma di controlli grafici, dipendenti dal tipo e dalla semantica della *UserOption* che rappresentano.² Il sistema non effettua alcuna azione in quanto il fatto di memorizzare le scelte fino al “submit” viene tenuto dalla form html stessa.

1.12 Open in New Window

1.12.1 Basic course

Si assume che il client stia interagendo con il tab relativo al *Chart* che vuole generare.

Il client fa click sul pulsante “Open In New Window”.

Il sistema esegue questi passi:

- apre il collegamento in una nuova pagina vuota
- invoca lo use case 1.10 per costruire il *Chart*
- visualizza il *Chart* nella nuova pagina

²inserire qua il mappaggio tra il tipo di *UserOption* e il controllo grafico che viene bindato.

Chapter 2

Gantt chart

Overall UML diagram

2.1 Make Left Column

2.1.1 Basic course

Il client prende un reference al *GanttChartGenerator* e domanda di creare la colonna di sinistra del diagramma.

Il sistema esegue questi passi:

1. costruisce una colonna di larghezza di dimensione **fissa**, che viene calcolata cdns.
2. Il sistema effettua una ricerca per calcolare l'insieme dei *Task* necessari da scrivere nella colonna.
3. Per ogni *Task* trovato si scrive il WBS identifier
 - (a) se *UserOptionChoises* contiene la scelta *TaskNameOption* allora accoda alla stringa scritta al punto prima il nome del task

2.1.2 Alternative course

troncamento dei TaskName se la stringa scritta supera la dimensione fissata nel documento di specifica, allora troncala secondo cdns.

troncamento del WBS identifier se avessi un identifier troppo lungo dovrei fare lo stesso??

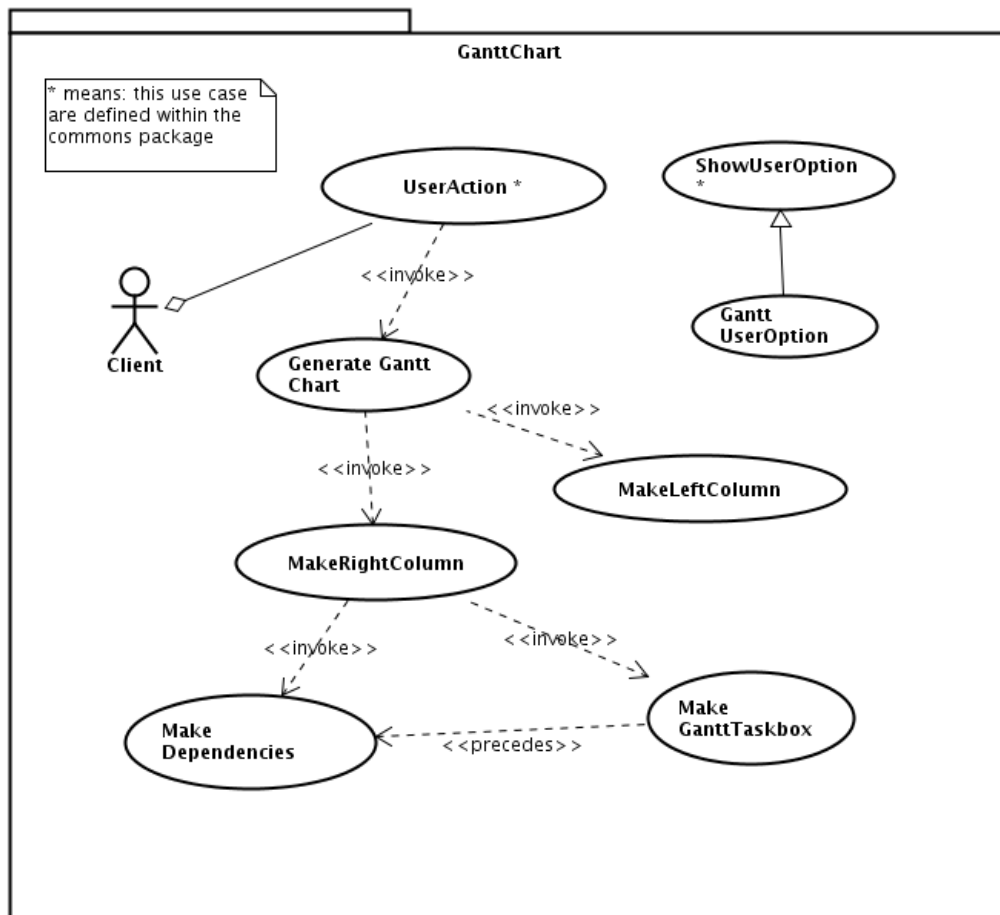


Figure 2.1: Gantt Overall UML diagram

2.2 Make GanttTaskbox

2.2.1 Basic course

Il client prende un reference al GanttChartGenerator e domanda di creare la rappresentazione grafica di un *Task*.

Il sistema esegue questi passi:

1. recupera il *Task* da rappresentare
2. costruisce la rappresentazione grafica, il *GanttTaskBox* in base alla scelta *WBSTreeSpecification*.
3. È possibile codificare delle informazioni aggiuntive:

- se *UserOptionChoice* contiene *ResourcesDetailsOption*, allora sul margine destro della *GanttTaskBox* appendi la stringa contenente la lista delle risorse cdns. Altrimenti appendi sul margine destro l'effort cdns.

2.2.2 Alternative course

troncamento delle resources se le informazioni testuali aggiuntive superano la dimensione definita nel documento di specifica, allora si troncano cdns.

2.3 Make Right Column

2.3.1 Basic course

Il client prende un reference al *GanttChartGenerator* e domanda di creare la colonna di destra del diagramma.

Il sistema esegue questi passi:

1. costruisce una colonna di larghezza di dimensione *fissa* che viene calcolata cdns.
2. Il sistema effettua una ricerca per calcolare l'insieme dei *Task* necessari da rappresentare nella colonna.
3. Per ogni *Task* trovato:
 - invoca lo use case 2.2
 - si posiziona il *GanttTaskBox* creato nella giusta posizione temporale in base alla scelta *TimeGrainOption*.
4. se *UserOptionsChoice* contiene *ShowDependencies* allora per ogni *TaskBox* rappresentata, invoca lo use case 1.1

2.3.2 Alternative course

troncamento delle resources se le informazioni testuali aggiuntive superano la dimensione definita nel documento di specifica, allora si troncano cdns.

2.4 Generate Gantt Chart

2.4.1 Basic course

Il client richiede di generare un *GanttChart*.

Il sistema costruisce un oggetto in questo modo:

- invoca lo use case 2.1 per creare la colonna a sinistra.
- invoca lo use case 2.3 per creare la rappresentazione nel tempo (colonna destra del diagramma).

Chapter 3

WBS chart

Overall UML diagram

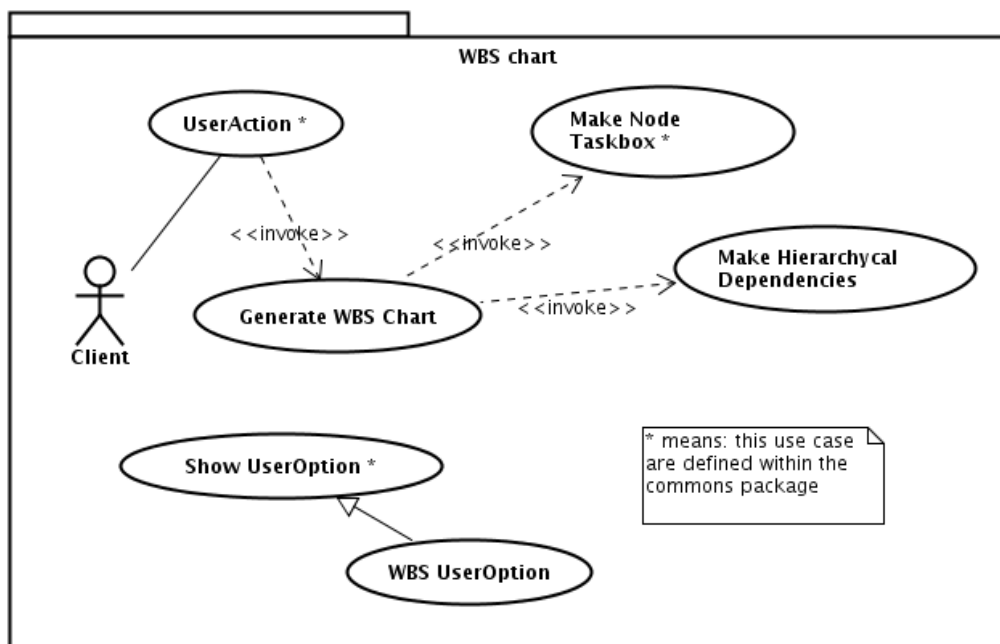


Figure 3.1: WBS UML diagram

3.1 Make Hierarchical Dependencies

3.1.1 Basic course

Il client richiede la funzionalità di rappresentazione della relazione gerarchica di un task.

Il sistema esegue questi passi:

1. in base alle informazioni in *WBSTreeSpecification*, recupera i figli del task (non l'intera discendenza, solo quelli di livello successivo) e li dispone graficamente¹ al di sotto di esso (top-down)²
2. Il sistema collega il task ai suoi figli con linee spezzate cdns.

3.1.2 Alternative course

WBSExplosionLevel = 0 se il livello di visualizzazione della *WBSStructure* richiesto si limita alla root, crea il solo un *NodeTaskbox* rappresentante l'intero progetto.

3.2 Generate WBS Chart

3.2.1 Basic course

Il client richiede di generare un *WBSChart*.

Il sistema costruisce un oggetto in questo modo:

- invoca lo use case 1.2 per creare i *NodeTaskBox* del diagramma.
- invoca lo use case 3.1 per creare le dipendenze gerarchiche

¹OSS. Non tutte le opzioni, in questa situazione hanno senso (resources detail?, alert marks?)

²effettivamente non e' che e' lui che li crea, dovrebbe solo aggiungerci le spezzate.

Chapter 4

TaskNetwork chart

Overall UML diagram

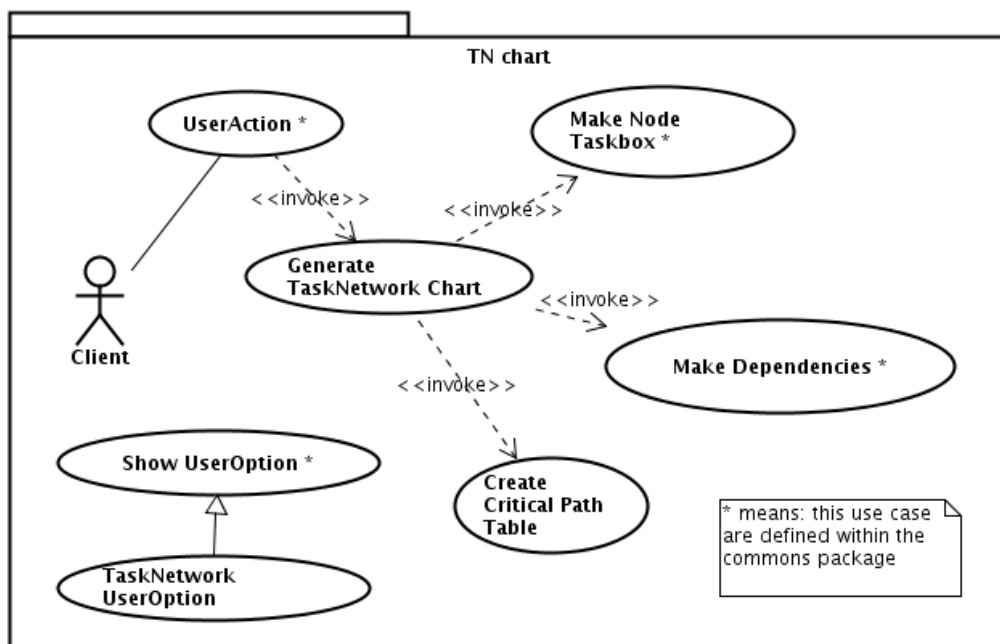


Figure 4.1: TaskNetwork UML diagram

4.1 Generate TaskNetwork Chart

4.1.1 Basic course

Il client richiede di generare un *TaskNetworkChart*.

Il sistema costruisce un oggetto in questo modo:

- invoca lo use case 1.2 per creare i *NodeTaskBox* del diagramma.
- si possono aggiungere le seguenti informazioni:
 - se *UserOptionsChoice* contiene *ShowCriticalPath* allora calcola il critical path rispetto al grafo rappresentato
 - se *UserOptionsChoice* contiene *ShowCriticalPathTable* allora invoca lo use case 4.2
 - se *UserOptionsChoice* contiene *ShowDependencies* allora per ogni *TaskBox* rappresentata, invoca lo use case 1.1

4.2 Create Critical Path Table

4.2.1 Basic course

Si assume che i critical path siano già stati calcolati e siano già rappresentati secondo le opzioni presenti in *UserOptionsChoice*.

Il client richiede di generare la tabella riassuntiva per i *critical path*.

Il sistema costruisce una tabella definita così: Ogni critical path calcolato rappresenta una

durata	effort	ultimo gap	visualizzato
...	*

entry nella tabella.