

Chapter 2

WBS

2.1 MakeChart

Il client prende un reference al *WBSChartGenerator* e domanda la funzionalità per creare il diagramma.

Il sistema costruisce il diagramma esploso al primo livello della WBS, di *font style*, *font size*, e *node_size* (calcolata in relazione a *font style* e *font size*) fissati (situazione di default).

2.1.1 Task Name Option

Il client richiede la funzionalità *Show Task Name*.

Il sistema nota che in *UserOptionChoises* è presente l'opzione *TaskNameOption*.

BASIC COURSE

Se *node_size* non è stata raddoppiata, $node_size = node_size * 2$.

Il sistema, $\forall task$ visualizzato, scrive $::(WBS_identifier(task), Get_TaskName(task))$.

ALTERNATIVE COURSES

Troncamento task name se $length::(WBS_identifier(task), Get_TaskName(task)) > node_size$ (2.1) allora il sistema chiamerà la funzione *cut* (1.4) sul *Task Name* che restituisce *cutted*.

Il sistema, $\forall task$ in cui si verifica (2.1), scrive $::(cutted, "...")$ (1.5).

Troncamento WBS id ??

2.1.2 Resources Detail Option

Il client richiede la funzionalità *Show Resources Detail*.

Il sistema nota che in *UserOptionChoises* è presente l'opzione *ResourcesDetailOption*.

BASIC COURSE

Se *node_size* non è stata raddoppiata, $node_size = node_size * 2$.

Il sistema controlla se è attiva la *UserOption AcutalDataOption*.

- Se non è attiva

Il sistema, $\forall task$ visualizzato, aggiunge al nodo presente una *Textual Strip* per ogni risorsa associata al task stesso.

Le *Textual Strip* saranno della forma $::(Get_PlannedResourceEffort(r), ",", Get_ReourceName(r), ",", Get_ResourceRole(r))$, \forall risorsa *r* assegnata al task.

Il sistema compone le *Textual Strip* risultanti all'interno di una unica *Strip* vuota (*Resource Detail*) della stessa larghezza del nodo, e di altezza pari all'altezza di n *Strip*, dove n è il numero di risorse r assegnate al task stesso.

- Se è attiva (procedura (1.1))

Il sistema, $\forall task$ visualizzato, aggiunge al nodo presente una *Textual Strip* per ogni risorsa associata al task stesso.

Le *Textual Strip* saranno della forma $::(Get_ActualResourceEffort, "/", Get_PlannedResourceEffort(r), "/", Get_ResourceName(r), ",", Get_ResourceRole(r)), \forall$ risorsa r assegnata al task.

Il sistema compone le *Textual Strip* risultanti all'interno di una unica *Strip* vuota (*Resource Detail*) della stessa larghezza del nodo, e di altezza pari all'altezza di n *Strip*, dove n è il numero di risorse r assegnate al task stesso.

ALTERNATIVE COURSES

Troncamento Textual Strip se $length(::(Get_PlannedResourceEffort(r), "/", Get_ResourceName(r), ",", Get_ResourceRole(r))) > node_size$ (2.2) ovvero se $length(::(Get_ActualResourceEffort, "/", Get_PlannedResourceEffort(r), "/", Get_ResourceName(r), ",", Get_ResourceRole(r))) > node_size$ (2.3), allora il sistema chiamerà la funzione *cut* (1.4) sulle *Textual Strip* di *Resource Detail* che restituisce *cutted*.

Il sistema, $\forall task$ in cui si verifica (2.2) ovvero (2.3), scrive $::(cutted, "...")$ (1.5).

2.1.2 Alert Marks User Option

Il client richiede la funzionalità *Show Alert Marks*.

Il sistema nota che in *UserOptionChoises* è presente l'opzione *AlertMarkUserOption*.

BASIC COURSE

Il sistema integra i nodi relativi ai soli task che presentano differenze tra i valori ACTUAL(Data Inizio, Data Fine, Budget) e i rispettivi valori PLANNED(Data Inizio, Data Fine, Budget) con un *Alert_Symbol*, inserito in un circolo centrato sull'angolo in alto a destra e di raggio pari alla metà dell'altezza di una *Strip*.

- *Alert_Symbol* Δ

Actual_value < *Planned_value*, per i valori Data Inizio, Data Fine, Budget.

- *Alert_Symbol* $\Delta!$

Actual_value > *Planned_value*, per i valori Data Inizio, Data Fine, Budget.

ALTERNATIVE COURSES