

UNIVERSITA' DEGLI STUDI DI PADOVA  
Corso di Ingegneria del Software 2016/2017

Appalto per la costruzione di un software per diagrammi UML

# SWEDesigner

# Oggetto dell' appalto

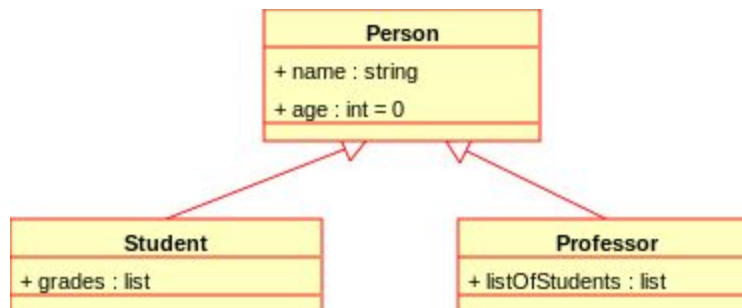
Il presente capitolato ha per oggetto l' affidamento della fornitura per la realizzazione di un software di costruzione di diagrammi UML con la relativa generazione di codice Java e Javascript.

L' innovazione oggi è la costante di qualunque settore di attività lavorativa. Motore dell' innovazione è il software, che permette di inserire elementi di agilità ed intelligenza in ogni attività umana, dalla fornitura di servizi alle realizzazioni meccaniche.

La costante richiesta di nuovo software si scontra con la cronica mancanza di esperti e con la bassa qualità del software prodotto; per aumentare la qualità e la velocità di produzione occorre rendere questa attività un processo industriale ingegnerizzato allontanandosi dall' artigianalità che ancora a volte lo caratterizza.

Lo standard per illustrare attraverso diagrammi le realizzazioni software è l' UML, Unified Modeling Language. Sono previsti vari tipi di diagrammi, a copertura di diverse fasi e di vari aspetti della costruzione dei prodotti software.

L' UML però non è esente da critiche, in particolare è molto debole nella sincronizzazione tra il codice e quanto disegnato nei diagrammi stessi.

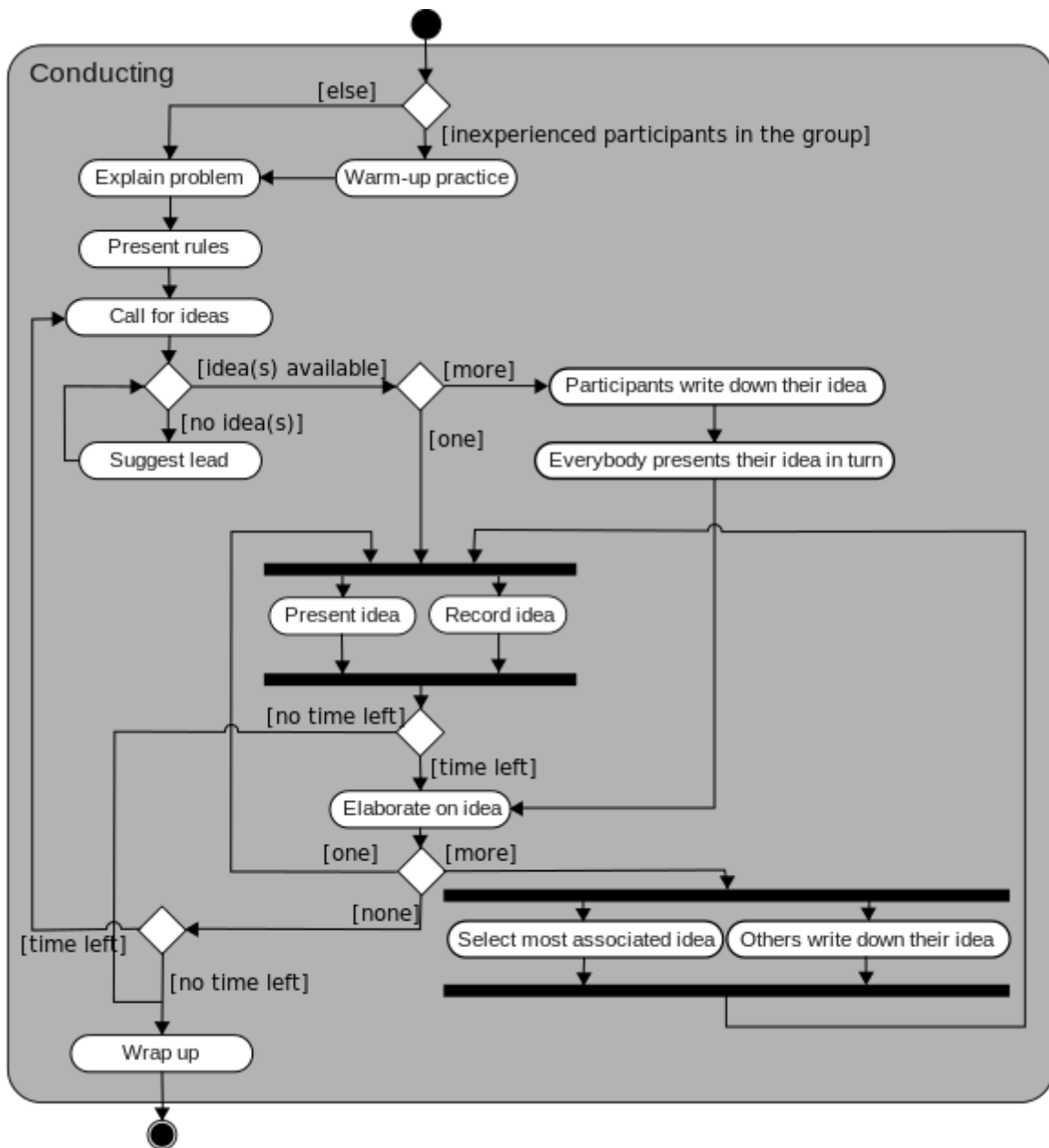


Con questo appalto l' azienda Zucchetti chiede di realizzare un disegnatore di diagrammi UML che sperimenti la possibilità di avvicinare le due fasi di progettazione e codifica, esplorando le interazioni e le eventuali estensioni necessarie a legare tra loro queste due fasi.

In particolare chiediamo di realizzare due tipi di diagrammi tra i tanti previsti dall' UML, il diagramma delle classi e il diagramma delle attività.

In entrambe i casi chiediamo che venga realizzato il codice espresso dal disegno, nel primo caso si tratterà dello scheletro delle classi, nel secondo caso del corpo dei metodi.

Non tutti i diagrammi delle classi e i diagrammi delle attività sono traducibili in codice sorgente, il sistema dovrà indicare quando questo è possibile e effettivamente realizzare il codice sorgente.



Probabilmente potrebbero essere di aiuto altri due diagrammi: il “sequence diagram” o anche il vecchio “flowchart” che non è previsto dall’ UML. Sarà a discrezione del gruppo che realizza il pacchetto esplorare l’ utilità (o la necessità per lo scopo prefissato) di questi ulteriori diagrammi.

# Caratteristiche e Requisiti Obbligatori

Il sistema che richiediamo sarà composto da almeno due disegnatori ed un coordinatore di quanto specificato dai disegnatori.

Il coordinatore governerà anche le fasi di costruzione del codice. I dati raccolti disegnando il progetto con i due tipi di diagrammi dovranno essere integrati per giungere alla realizzazione del codice sorgente completo: scheletro delle classi e corpo dei metodi.

Il sistema dovrà essere in grado di produrre codice nei due linguaggi di programmazione Java e Javascript.

Il software richiesto dovrà svolgere almeno i seguenti compiti:

1. Realizzare diagrammi delle classi
2. Realizzare diagrammi delle attività
3. Produrre lo scheletro delle classi in Java o Javascript partendo dal diagramma delle classi
4. Produrre il corpo dei metodi in Java o Javascript partendo dal diagramma delle attività
5. Associare un diagramma delle attività alla definizione di metodo nel diagramma delle classi
6. Indicare se i diagrammi contengono elementi per cui non è possibile generare il codice perché incompleti o non realizzabili nel linguaggio target prescelto
7. Rendere disponibili le informazioni specificate nel diagramma associato (es: attributi e metodi della classe mentre viene disegnato il diagramma di attività per un metodo)

Il sistema dovrà essere realizzato con tecnologie Web. In particolare richiediamo che la parte server venga realizzata in Java con server Tomcat o Javascript con server Node.js. La parte client dovrà essere eseguibile in un browser HTML5 ed utilizzare fogli stile CSS per l'aspetto estetico e Javascript per la parte attiva.

## Requisiti Opzionali

Il software potrebbe avere le seguenti caratteristiche:

1. Realizzare anche altri diagrammi previsti dall' UML, in particolare per questo compito potrebbe essere utile il "sequence diagram"
2. Realizzare anche altri diagrammi non previsti dall' UML, in particolare per questo compito potrebbe essere utile il "flowchart"

3. Per l'organizzazione di progetti di una certa dimensione è utile il "package diagram", opzionalmente potrebbe essere realizzato il diagramma specifico o può essere introdotta la gestione dei package nel disegnatore delle classi
4. Proporre estensioni o modifiche ai diagrammi implementati al fine di rendere più efficiente la generazione del codice.
5. Poiché i sistemi automatici non sempre riescono a produrre il 100% del programma, realizzare un sistema che permetta la realizzazione di piccole modifiche al sorgente generato, mantenendole nelle generazioni successive.
6. Nel produrre codice i "Patterns" sono la forma più rapida per riusare soluzioni di comprovata validità. Introdurre nei disegnatori la possibilità di riutilizzare un pattern descritto in precedenza.

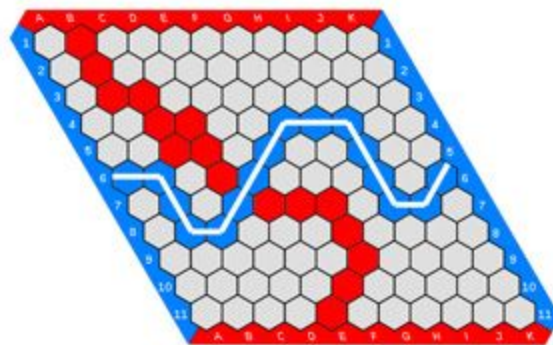
A titolo di ispirazione ed esempio segnaliamo l'esistenza di parecchi software, anche open-source, che trattano temi simili.

Nell'ambiente desktop indichiamo ArgoUML, StarUML, Software Idea Designer, UMLet e PlantUML.

In ambiente web parecchi disegnatori permettono la costruzione di diagrammi UML, come ad esempio LucidChart.

Come ulteriore requisito, da considerarsi più un "desiderata" che un requisito opzionale, chiediamo di realizzare un programma di una certa consistenza utilizzando lo strumento stesso.

Per questo tema proponiamo la realizzazione di un programma che sia in grado di giocare al gioco "Hex", inventato indipendentemente da Piet Hein e John Nash.



La realizzazione di questo gioco è abbastanza semplice da poter essere realizzata anche con uno strumento incompleto, mentre può essere di grande aiuto per verificare il livello raggiunto dal sistema di generazione del codice.

La programmazione della scacchiera e della gestione della partita tra due giocatori umani è relativamente semplice, invece chi si appassionasse a questo affascinante gioco potrebbe cimentarsi anche nella realizzazione dell'Intelligenza Artificiale per l'avversario computerizzato, compito decisamente di livello superiore.

## Variazioni ai requisiti

In corso d'opera non sarà possibile variare/modificare i requisiti minimi (obbligatori per accettare il prodotto). Sarà invece possibile variare i requisiti opzionali, in quanto saranno i gruppi vincitori dell'appalto a modificarli / eliminarli / aggiungerli.

## Documentazione

Il progetto dovrà essere supportato dalla documentazione minima richiesta per il corso di Ingegneria del software e dovrà essere fornito un manuale per l'utilizzo ed un manuale per chiunque voglia estendere l'applicazione.

## Garanzia e Manutenzione

L'azienda Zucchetti SPA è interessata a questo progetto come dimostrazione della fattibilità dell'obiettivo utilizzando le tecnologie web. Costituirà titolo preferenziale nella valutazione delle proposte la pubblicazione del progetto sul sito "github.com" o altri repository pubblici, in conformità con i relativi requisiti di natura open-source, per favorire la continuità del prodotto risultante.

## Rinvio

Per tutto quanto non previsto nel presente capitolato, sono applicabili le disposizioni contenute nelle leggi e nei collegati per la gestione degli appalti pubblici.