

# Requisiti soddisfatti

## 1 Introduzione

In questo documento si riportano le funzionalità che sono state implementate, con eventuali considerazioni, in stretta relazione con il documento dei requisiti. Si è scelto di usare i colori per rendere più evidente i risultati raggiunti punto per punto:

- ✓ Il verde indica un obiettivo pienamente raggiunto, che non ha presentato problemi in fase di test.
- ! L'arancione indica un obiettivo in gran parte raggiunto, ma con qualche riserva dettagliata puntualmente
- X Il rosso indica un obiettivo raggiunto solo parzialmente.

## 2 Requisiti

### 2.1 Requisiti funzionali

#### 2.1.1 Obiettivi primari

- 1 L'interfaccia grafica del plugin deve permettere l'accesso a tutte le funzionalità del tool TwoTowers versione 5.1 già presente elencate di seguito:

1.1 Operazioni standard su file che elabora il programma, quali: ✓

1.1.1 visualizzazione e modifica; ✓

1.1.2 creazione; ✓

1.1.3 caricamento; ✓

1.1.4 salvataggio con nome; ✓

1.1.5 salvataggio (senza un nome nuovo); ✓

Tutte funzionalità già presenti in Eclipse. Anche gli editor personalizzati (per ciascun tipo di file) necessari per la visualizzazione e modifica dei file usano le funzionalità di base messe a disposizione dal framework di Eclipse. Per questo sono funzionalità altamente affidabili.

## 1.2 Aemilia COMPILER ✓

- Parser ✓
- Integrated semantic model size calculator ✓
- Functional semantic model size calculator ✓
- Performance semantic model size calculator ✓
- Integrated semantic model generator ✓
- Functional semantic model generator ✓
- Performance semantic model generator ✓

Tutte funzioni fedelmente riportate dal precedente tool e funzionanti. Sono accessibili sia dal menù principale, sia dalla toolbar e si attivano/compaiono solo quando è aperto con un editor un file su cui è possibile fare tali operazioni.

## 1.3 EQUIVALENCE VERIFIER ✓

- Strong bisimulation equivalence verifier ✓
- Weak bisimulation equivalence verifier ✓
- Strong Markovian bisimulation equivalence verifier ✓
- Weak Markovian bisimulation equivalence verifier ✓

Tutte funzioni fedelmente riportate dal precedente tool e funzionanti. Sono accessibili sia dal menù principale, sia dalla toolbar e si attivano/compaiono solo quando è aperto con un editor un file su cui è possibile fare tali operazioni.

#### 1.4 MODEL CHECKER !

- Symbolic LTL model checker !

Funzione fedelmente riportata dal precedente tool. E' accessibile sia dal menù principale, sia dalla toolbar e si attiva/compare solo quando è aperto con un editor un file su cui è possibile fare tale operazione. La funzionalità è stata scarsamente testata a causa di problemi con il tool NuSMV indipendenti dalla realizzazione del plugin e da Eclipse. Essendo l'implementazione del tutto analoga a quella delle altre funzionalità fin qui presentate, è ragionevole pensare che funzioni correttamente in presenza del tool NuSMV installato.

#### 1.5 SECURITY ANALYZER ✓

- Non-interference analyzer ✓
- Non-deducibility on composition analyzer ✓

Tutte funzioni fedelmente riportate dal precedente tool e funzionanti. Sono accessibili sia dal menù principale, sia dalla toolbar e si attivano/compaiono solo quando è aperto con un editor un file su cui è possibile fare tali operazioni.

#### 1.6 PERFORMANCE EVALUATOR ✓

- Stationary probability distribution calculator (Gaussian Elimination) ✓
- Stationary probability distribution calculator (adaptive symmetric SOR) ✓
- Transient probability distribution calculator (uniformization) ✓
- Stationary reward-based measure calculator (Gaussian Elimination) ✓
- Stationary reward-based measure calculator (adaptive symmetric SOR) ✓
- Transient reward-based measure calculator (uniformization) ✓
- Simulator ✓

Tutte funzioni fedelmente riportate dal precedente tool e funzionanti. Sono accessibili sia dal menù principale, sia dalla toolbar e

si attivano/compaiono solo quando è aperto con un editor un file su cui è possibile fare tali operazioni.

2 Inoltre devono essere aggiunti i seguenti menù e sottomenù:

2.1 Architectural Assistant: ✓

- Compatibility Checker ✓
- Interoperability Checker ✓
- Queueing Network Generator ✓

Funzionalità già presenti nel menù principale, ma non abilitate. E' già predisposta la struttura sottostante (command e handler) pronta per accogliere un'implementazione rapida e analoga a quella delle altre funzioni.

2.2 Code Generator: ✓

- Program Generator ✓
- Package Generator ✓
- Applet Generator ✓

Funzionalità già presenti nel menù principale, ma non abilitate. E' già predisposta la struttura sottostante (command e handler) pronta per accogliere un'implementazione rapida e analoga a quella delle altre funzioni.

3 Integrazione dell'interfaccia realizzata con un tool esistente per la trasformazione di diagrammi UML in specifiche Aemilia. ✓ La funzionalità è presente sia nel menù principale che nella toolbar e consiste nell'esecuzione del programma Seal.

4 Stesura di un manuale utente, ovvero di un menù Help. ✓ E' stato adattato il manuale già presente nel tool TwoTowers, lasciando i contenuti inalterati, al sistema di Help presente in Eclipse.

5 Codice sorgente commentato e documentato attendendosi il più possibile allo stile di documentazione standard per i plugin di Eclipse. ! Non è stato possibile commentare il codice totalmente. Le motivazioni sono di natura tecnica e riguardano la natura particolare di un progetto di questo tipo.

### 2.1.2 Obiettivi secondari

- 1 Funzionalità di editing dei file avanzate per ciascun tipo di estensione usato:
  - syntax highlighting; **X**
  - content assist (per i file modificabili da utente). **X**

E' stato implementato il syntax highlighting per alcuni editor prendendo le parole chiave dal manuale. Il content assist invece non è attualmente presente, anche se è già predisposta la struttura delle classi. Quindi si rimanda alla documentazione ufficiale di Eclipse come punto di partenza per l'implementazione: Platform Plug-in Developer Guide > Programmer's Guide > Editors.

## 2.2 Requisiti non funzionali

Interfaccia grafica:

- 1 intuitiva; ✓ La gui è stata progettata sulla base di quella che usa Eclipse, tool ormai affermato e maturo. Si ritiene perciò che anche questo requisito sia pienamente raggiunto.
- 2 di facile navigabilità; ✓ Vale lo stesso discorso del punto precedente. Si sono usati meccanismi per organizzare le varie funzionalità in maniera analoga con ciò che già mette a disposizione Eclipse.
- 3 estendibile. ✓ Il framework di Eclipse è basato su componenti estendibili per definizione: i plugin. Risulta quindi estremamente facile estendere funzionalità ai plugin implementati, o aggiungerne di nuovi più o meno dipendenti da questi.