

d-asap @Unibg



Kickoff meeting
Milano 29 gennaio 2009

Componenti gruppo

- **Fa parte dell'unità del PoliMi**
- **Membro**
 - Angelo Gargantini

UniBg Goal Principale

- in WP4: **Tecniche e strumenti per il testing** (leader CNR)
- *si intende integrare notazioni per la modellazione con tecniche e strumenti per il testing, basato su modelli, di applicazioni dinamiche affidabili*
- **Keywords:** model-based testing, combinatorial testing, test generation, tests and proofs integration
- Estendere/applicare quello fatto fino ad ora per reactive systems / embedded systems a dasap

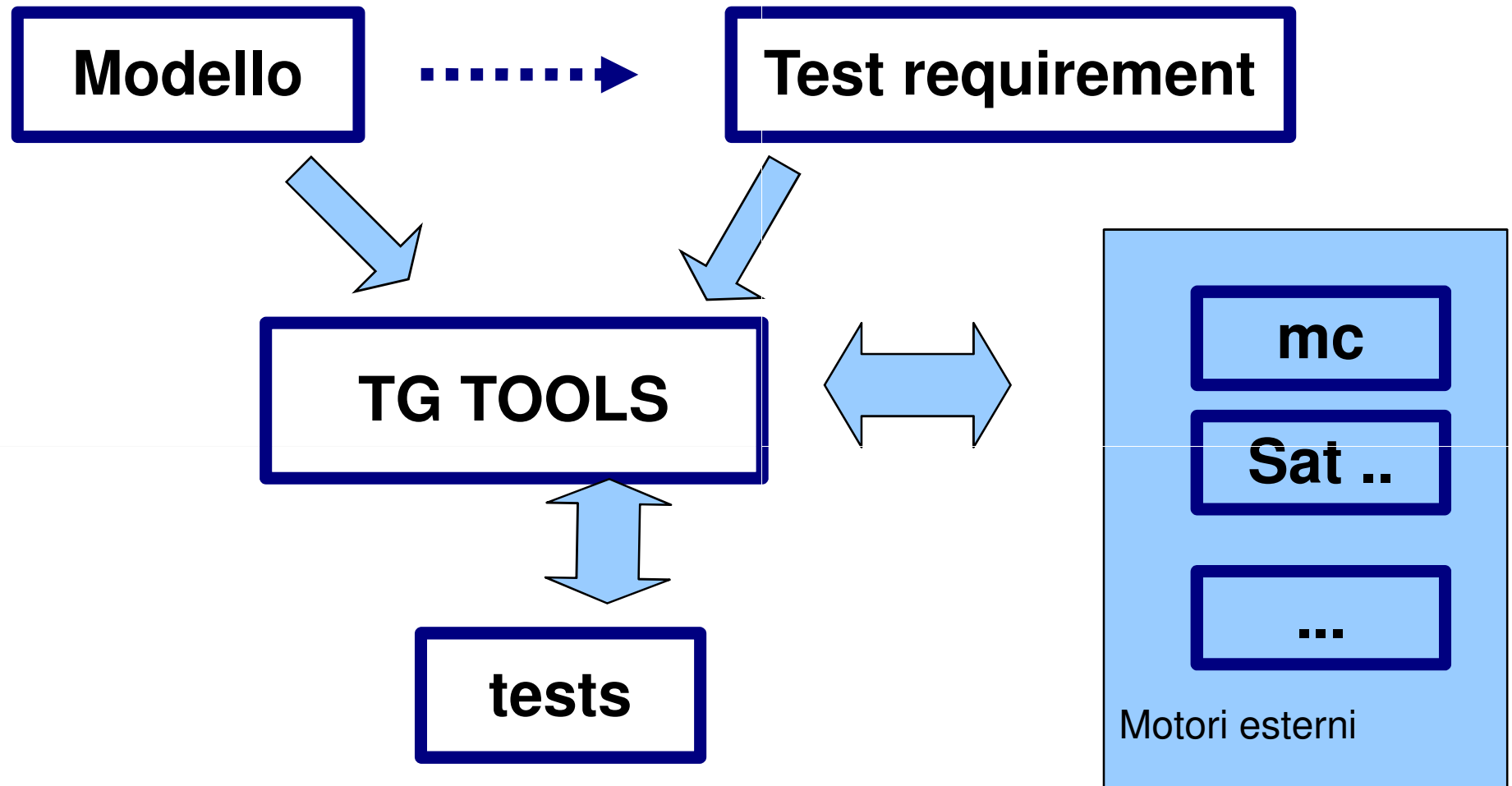
Stato dell'arte – ambiti di ricerca

- **Generazione di casi di test mediante model checking**
 - Angelo Gargantini, Constance L. Heitmeyer: Using Model Checking to Generate Tests from Requirements Specifications. ESEC/SIGSOFT FSE 99
 - Angelo Gargantini, Elvinia Riccobene: ASM-Based Testing: Coverage Criteria and Automatic Test Sequence. J. UCS 00
 - Gordon Fraser and Angelo Gargantini, Evaluation of Model Checkers for Test Case Generation, ICST 09
- **Combinatorial testing in presenza di constraints**
 - Andrea Calvagna, Angelo Gargantini: A Logic-Based Approach to Combinatorial Testing with Constraints. TAP 2008: 66-83
 - Andrea Calvagna and Angelo Gargantini, Using SRI SAL Model Checker for Combinatorial Tests Generation in the Presence of Temporal Constraints, AFM08
- **Generazione di scenari per l'animazione/validazione**
 - Angelo Gargantini, Elvinia Riccobene: Automatic Model Driven Animation of SCR Specifications. FASE 2003
 - A. Gargantini, E. Riccobene, P. Scandurra: Model-driven system validation by scenarios in Languages for Embedded Systems and their Applications. Lecture Notes in Electrical Engineering (LNEE) 2009

Stato dell'arte - tools

- Una “famiglia” di tools per la generazione di casi di test
 - Utilizzando diverse notazioni di ingresso: SCR/AsmGofer/AsmetaL
 - Utilizzando diverse tecniche: model checking simbolici/explicit state (Spin, smvs, sal), BMC/SAT, SMT (yices, hysat)
 - Supporta diversi criteri di copertura (structural, MCDC, fault -mutation, combinatorial interaction)
 - **Work in progress**: piattaforma estensibile in cui innestare i diversi componenti come plugin (eclipse)
 - Punto di forza: integra molte tecniche e si presta per esperimenti di confronto
 - <http://cs.unibg.it/gargantini/software>

Idea base [1999]



ASSUNZIONI

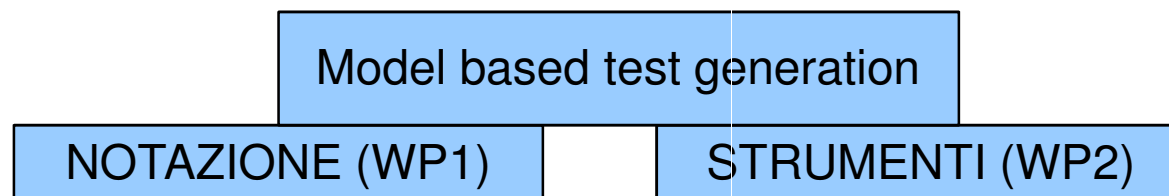
- Il problema può essere rappresentato mediante logica XXX
- Il motore riesce a risolvere (ad esempio mediante generazione di cex)

Problemi aperti

- Estrarre i test requirements più significativi
- Ordinamento del test requirements
- Aggregazione dei test req.
 - Ad esempio per minimizzare il numero di test (o la lunghezza,...)
- Trovare il motore più adatto
 - Ad esempio usare dMC

Integrazione con altri WP/unità

- **Notazioni:** si prevede di adattare l'approccio per modelli nelle notazioni sviluppate in WP1
 - adottate dal gruppo UniPi, ArchiTRIO o altri DSML sviluppate da altri gruppi
- **Strumenti:** si propone di utilizzare tecniche e tool sviluppati di verifica e/o altri model checker (Bogor, SPIN o SAT based) sviluppati nel WP2



Testing per Architetture Software Adattabili e Affidabili e per Sistemi Pervasivi

- Alcune linee di sviluppo
 - 1) Conformance testing
 - 2) Combinatorial testing per applicazioni web
 - 3) Generazione di scenari
- Liberamente prese da:
 - Bertolino, A.; Inverardi, P. & Muccini, H. Bernardo, M. & Inverardi, P. (ed.) Formal Methods in Testing Software Architectures SFM, Springer, 2003, 2804, 122-147
 - Gerardo Canfora, M. Di Penta. Service Oriented Architectures Testing: A Survey G ISSSE 2006-2008, LNCS 5413, A. De Lucia and F. Ferrucci Eds., pp. 78-105, 2009, Springer, 2009

1. Conformance testing

- *Test di conformità a livello di architetture software*
- derivazione di scenari critici (e possibilmente la loro animazione) per validare i modelli e riconoscere i difetti il prima possibile
- derivazione automatica di casi di test a partire dalle specifiche (da salvare in formato interscambiabile)
- tracciabilità: da scenari astratti ad esecuzioni concrete (come in ALTS)
- test execution in presenza di non determinismo

2. Combinatorial Interaction Testing in the presence of Constraints

Applicare il CCIT a architetture e/o ad applicazioni web (web services ...)

- Scoperta di modelli semplificati delle interfacce di web services (ad esempio http form, da spec da WP1, dal WSDL, ...),
- **Scoperta/ricerca di vincoli** (constraints)
- Generazione di casi di test per il CCIT per il test di conformità/performance/fault tolerance ...

Related work

Combinatorial for web applications

- Mao, Chengying, Performing Combinatorial Testing on Web Service-Based Software, Computer Science and Software Engineering, 2008 International Conference on
- Lei Xu, Baowen Xu and Changhai, Testing and Fault Diagnosis for Web Application Compatibility Based on Combinatorial Method, Workshop 8: International Workshop on Web Information Systems and Applications (WISA 2005)
- ...

Scoperta di modelli per web applications:

- W. Halfond and A. Orso, Improving Test Case Generation for Web Applications Using Automated Interface Discovery, Foundations of Software Engineering (ESEC/FSE 2007).

3. Generazione di scenari

- *Generazione di scenari critici che evidenziano comportamenti attesi in situazioni particolari*
- Tali scenari potranno essere utilizzati durante la fase di progettazione per validare i modelli alla luce di proprietà espresse formalmente, durante la fase di sviluppo
- Grazie ai test di conformità, e durante la fase di esecuzione per testare nuovi componenti che potrebbero essere aggiunti dinamicamente.