

## Abstract State Machines

Corso per il Dottorato di Ricerca in  
Informatica  
XXVII Ciclo

## Docente

- Alessandro Bianchi
  - Dipartimento di Informatica – V piano
  - Tel. 080 544 2283
  - E-mail [bianchi@di.uniba.it](mailto:bianchi@di.uniba.it)
  - Orario di ricevimento:
    - mercoledì 15:30 - 17:30
    - su appuntamento
  - URL <http://www.di.uniba.it/~bianchi/>

## Il Contesto (1)

- Caratteristica emergente nei sistemi informatici complessi è quella della **distribuzione** e **cooperazione** tra diversi agenti computazionali allo scopo di fornire servizi sempre più *raffinati*
  - La realizzazione di sistemi informatici complessi si sta configurando come **integrazione di moduli indipendenti**, spesso **distribuiti**, ciascuno capace di offrire servizi computazionali in **cooperazione** con gli altri

## Il Contesto (2)

- Modifica di paradigma: dalla **computazione** alla **comunicazione**
  - Necessità di **capire** e **modellare** opportunamente il comportamento di sistemi **di comunicazione**, analogamente alla comprensione che si è sviluppata riguardo i sistemi **di computazione**
- La comunicazione è vista come parte integrante della computazione

## Esempi di Sistemi Distribuiti Critici

- Mobile Ad-hoc NETWORKS – MANET
  - Reti wireless di computer che NON necessitano di infrastruttura fisica fissa
- Sistemi Grid
  - Insieme di risorse computazionali distribuite
- Sistemi per il file sharing
  - BitTorrent
- Protocolli di sicurezza distribuiti

## Motivazioni

- Necessità di trattare **formalmente** la comunicazione
  - tra i processi diversi di una stessa applicazione
  - all'interno di un processo
- Il formalismo è necessario per poter garantire la **Qualità del Servizio (QoS)**
- Formalismi per
  - Specificare
  - Analizzare
  - Verificare

## Modellazione (1)

- La realizzazione di sw richiede successive attività di **modellazione** per passare da una rappresentazione del problema da livelli di astrazione più alti a livelli più bassi
- Esistono diversi tipi di modellizzazioni
  - Informali
  - Semiformali
  - Formali

## Modellazione (2)

- I modelli non **devono** e non **possono** rappresentare tutto il sistema
  - **Separazione degli aspetti** il più possibile ortogonali fra loro
  - Si deve essere consci del fatto che gli aspetti da cui il modello astrae possono avere effetti che NON possono essere considerati
- Per ogni aspetto di interesse si definisce un modello che:
  - lo rappresenti come concetto “chiave”
  - che astragga da altri aspetti meno importanti

## Modellazione (3)

- Per risolvere un problema estremamente complesso
  - lo si divide in diversi livelli di astrazione, affrontati in sequenza, ad esempio: top-down, bottom-up, o combinati
- La scelta del modello è essenziale per poter affrontare problemi complessi e per la qualità della realizzazione
  - Eventuali failure possono avere effetti disastrosi
  - La sicurezza deve essere garantita
  - ...

## Modellazione (4)

- È spesso necessario adottare modellizzazioni che favoriscano la validazione dell'applicazione rispetto ai parametri di qualità desiderati

## Analisi di proprietà computazionalmente interessanti

- L'adozione di modelli formali facilita l'analisi di proprietà particolarmente interessanti dal punto di vista computazionale
  - Deadlock/Livelock
  - Starvation/Liveness
  - Reachability
  - Reversibility
  - ...

## Formalizzazione

- Permette di creare una specifica più completa, uniforme e non ambigua rispetto a quanto ottenuto con altri metodi
- Della specifica formale può essere dimostrata la correttezza
- È spesso necessario adottare modellizzazioni che favoriscano la validazione dell'applicazione rispetto ai parametri di qualità desiderati

## Svantaggio della formalizzazione

- Adottare metodi di sviluppo formali:
  - è difficile
  - è impegnativo
  - richiede molto tempo
  - richiede elevate competenze

## Il Metodo ASM

- È un metodo per lo sviluppo di sistemi sw complessi, basato sul formalismo delle Abstract State Machine
- Delega a chi svolge la modellazione la profondità del formalismo

## Obiettivi del Corso

- **Fornire**
  - conoscenze sui concetti fondativi del formalismo delle ASM
  - capacità di applicare il metodo di sviluppo ASM-based
- **Stimolare**
  - analisi critica delle conoscenze acquisite

## Prerequisiti e Caratteristiche Richieste

- Conoscenze di base fornite nei corsi di Informatica triennale
- Capacità di astrazione e formalizzazione
- Desiderio di applicare le conoscenze per indagare fenomeni che si presentano in pratica

## Programma (1)

- Introduzione alle Abstract State Machine (ASM)
  - Il problema
  - Scopo e caratteristiche
  - Applicabilità
- Concetti di base
  - Richiami sulle FSM
  - Da FSM a ASM
  - Il formalismo delle ASM

## Programma (2)

- ASM Bohem-Jacopini
  - Costrutti di sequenza-selezione-iterazione
  - Parametrizzazione
- Il metodo ASM-based
  - Ground model
  - Raffinamenti verticali
  - Raffinamenti orizzontali
- Introduzione a Distributed ASM (DASM)
  - sync\_DASM
  - async\_DASM

## Programma (3)

- Analisi delle Proprietà
- Esempi di applicazione
  - Mobile Ad-hoc NETWORK (MANET)
  - Grid Job Management
  - ...

## Valutazione

- Scopo della valutazione: verificare
  - l'apprendimento dei concetti
  - le capacità di applicarli per risolvere problemi specifici
- Seminario monografico, oppure svolgimento di un caso di studio

# Bibliografia

- Testo
  - E. Börger, R. Stärk, *Abstract State Machine*, Springer 2003
- Lucidi del corso, disponibili a partire dal sito
  - [http://www.di.uniba.it/~bianchi/didattica/2011\\_12/asm/index.htm](http://www.di.uniba.it/~bianchi/didattica/2011_12/asm/index.htm)
- Ulteriori riferimenti
  - Articoli e lucidi citati / distribuiti durante le lezioni