DIETI UNINA UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

Corso di Laurea in Ingegneria Informatica

Corso di Ingegneria del Software

Model-based Testing

Sommario

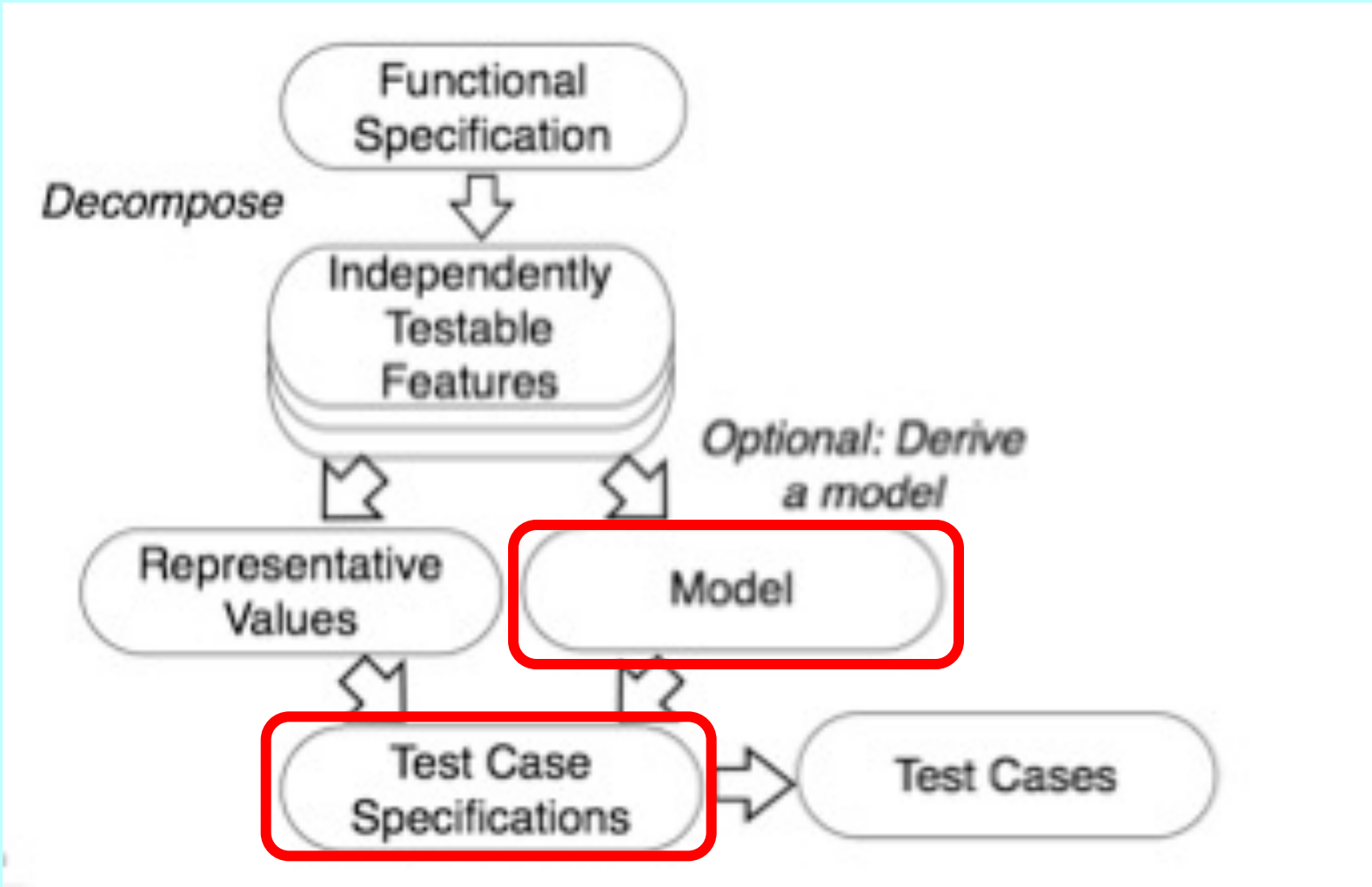
- Model-based testing: terminologia
- Criteri di copertura
- Benefici e limiti del model-based testing

Riferimenti:

M.Pezzè, M. Young; Software Testing and Analysis. Process, principles, and techniques.
Cap. 14 § § , 14.1, 14.2

Harry Robinson Tutorial – Google. Slide disponibili al sito:
www.model-based-testing.org

Dalla specifica ai casi di test



Pezzè & Young, Software Testing And Analysis: Process, Principles And Techniques, 2008

Approcci al test funzionale

- Data una specifica ci possono essere più approcci per derivare i casi di test

- Oltre alle tecniche di tipo combinatoriale (**category-partition** e **pairwise testing**), le **specifiche di casi di test** (da cui si generano i casi di test effettivi) possono essere derivate tramite un **modello**

Model-based Testing

- Cos'è un modello?

- Un modello è una descrizione "conveniente" di un sistema
- I modelli sono più semplici dei sistemi che descrivono
- I modelli aiutano a comprendere e predire il comportamento del sistema

Cos'è il model-based testing?

- *"Il model-based testing è una tecnica in cui il comportamento a tempo d'esecuzione di una implementazione sotto test viene verificato rispetto alle predizioni fatte da una specifica formale, o modello."* - Colin Campbell, MSR
- In altri termini, un modello descrive come un sistema dovrebbe comportarsi in risposta ad una azione
- Stimolare il sistema con un'azione e vedere se il sistema risponde come atteso

Model-based Testing

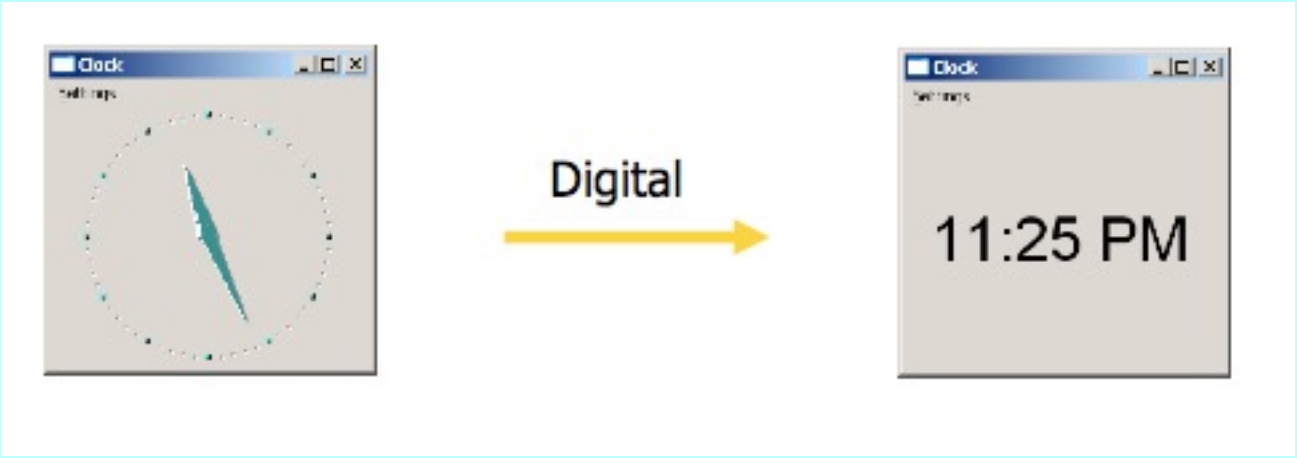
- Le tecniche di test strutturale basate sul *Control flow* e sul *data flow*, sono comunque basate su modelli. I modelli sono estratti dal codice sorgente.
- I modelli possono essere estratti dalle specifiche o dal design, permettendo di sfruttare informazioni riguardo il **comportamento atteso**
- *Il model-based testing consiste nel derivare modelli del comportamento atteso per produrre i casi di test*

Usare i modelli per testare

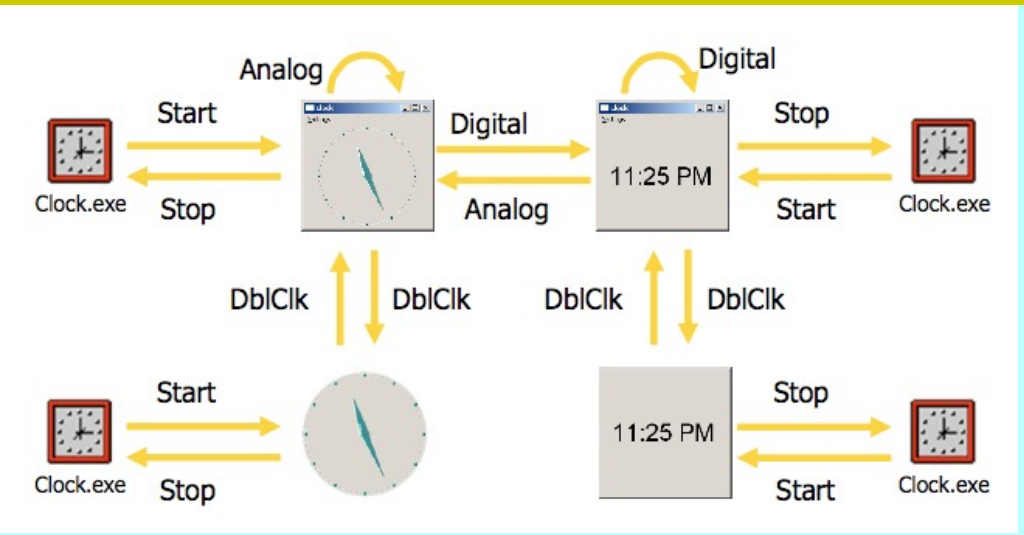
- Che tipo di modello uso?
- Come creo un modello?
- Come scelgo i test?
- Come verifico i risultati?

- Vi sono diversi tipi di modelli
  - *Macchine a stati, Insiemi, Grammatiche, Combinazioni, ecc.*

Esempio: modello dell'orologio



Possiamo sostiuire questo modello…



- … Con un modello a stati

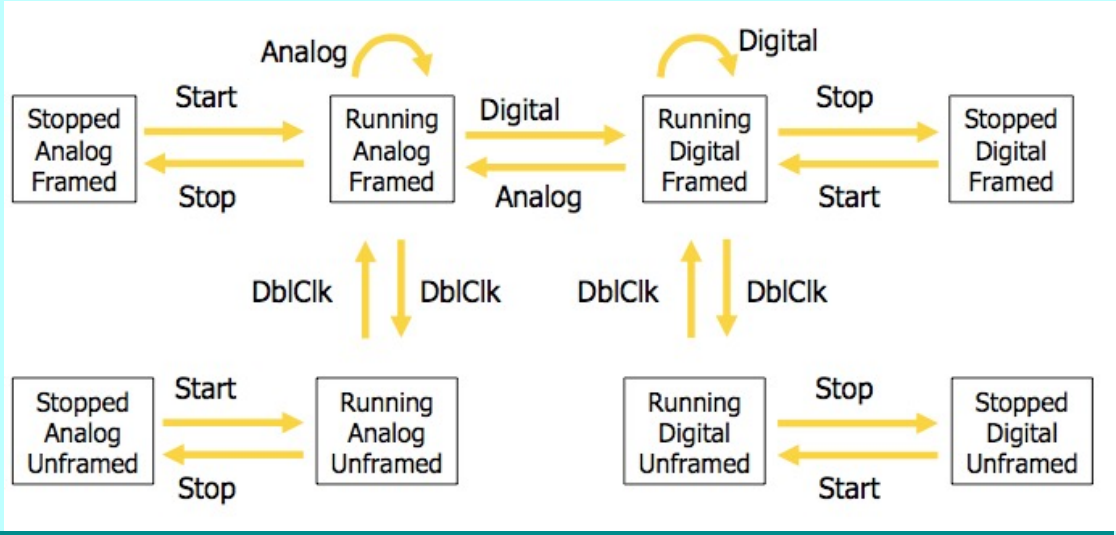
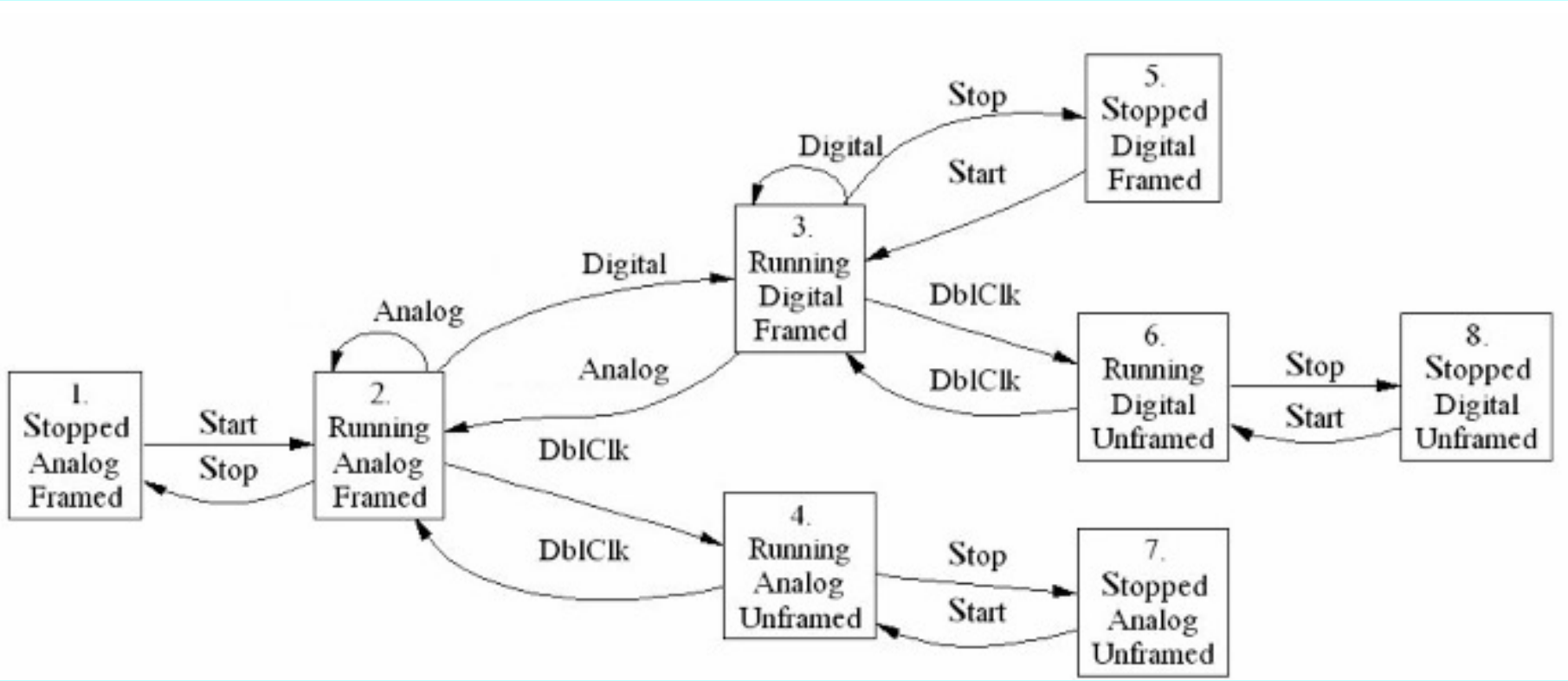


Tabella degli stati

STARTSTATE	ACTION	ENDSTATE
Stopped Analog Framed	Start	Running Analog Framed
Running Analog Framed	Stop	Stopped Analog Framed
Running Analog Framed	SelectAnalog	Running Analog Framed
Running Analog Framed	SelectDigital	Running Digital Framed
Running Analog Framed	DblClk	Running Analog Unframed
Running Digital Framed	Stop	Stopped Digital Framed
Running Digital Framed	SelectAnalog	Running Analog Framed
Running Digital Framed	SelectDigital	Running Digital Framed
Running Digital Framed	DblClk	Running Digital Unframed
Running Analog Unframed	Stop	Stopped Analog Unframed
Running Analog Unframed	DblClk	Running Analog Framed
Stopped Digital Framed	Start	Running Digital Framed
Running Digital Unframed	Stop	Stopped Digital Unframed
Running Digital Unframed	DblClk	Running Digital Framed
Stopped Analog Unframed	Start	Running Analog Unframed
Stopped Digital Unframed	Start	Running Digital Unframed

State diagram



Generazione dei test

- Possiamo adottare diversi criteri per generare le sequenze dei casi di test a partire dal modello:
  - Random walk (simile al *random testing*)
  - All transitions (copertura delle transizioni)
  - All states (copertura degli stati)
  - Shortest paths first
  - Most likely paths first

Una sequenza *all-transitions*

1. Start
2. Analog
3. Digital
4. Digital
5. Stop
6. Start
7. Double Click
8. Stop
9. Start
10. Double Click
11. Analog
12. Double Click
13. Stop
14. Start
15. Double Click
16. Stop

Perchè il model-based testing funziona?

- *"…I think that less than 10 percent of most programs' code is specific to the application. Furthermore, that 10 percent is often the easiest 10 percent. Therefore, it is not unreasonable to build a model program to use as an oracle."*

–Boris Beizer, Black Box Testing, p.63

Benefici del model-based testing

- Casi di test facili da manutenere
- Generazione dei casi di test semplice => più casi di test a parità di tempo => costi ridotti
- La parte più impegnativa (che richiede sforzo umano) è la creazione di un modello: se si ha già a disposizione un modello dalle specifiche, è conveniente sfruttarlo per generare i casi di test

Problemi del Model-Based Testing

- Richiede competenze
  - Per la fase di modellazione
- **Richiede test di conformità (*conformance testing*)**
  - Pur testando il sistema rispetto ad un modello, è necessario assicurare che il modello sia corretto, i.e., che catturi effettivamente il comportamento del sistema
- Un modello esaustivo del sistema può richiedere risorse ingenti
  - In questo caso si può adottare questa tecnica per una parte del sistema

Problemi del Model-Based Testing

- Metriche
  - Quali metriche? Numero di casi di test, copertura delle transizioni o degli stati …
  - La copertura delle specifiche o del codice (nel test funzionale e strutturale, rispettivamente) sono metriche migliori