Szoftverellenőrzési technikák

Tesztelés a fejlesztés különböző fázisaiban

Majzik István, Micskei Zoltán

http://www.inf.mit.bme.hu/

1

Utolsó módosítás: 2012.10.29.

Tartalom

- Modul / Unit tesztelés (korábbi előadás)
- Integrációs tesztelés
- Rendszer tesztelés
- Elfogadás tesztelés

- Tesztek leírása: U2TP

Integrációs tesztelés

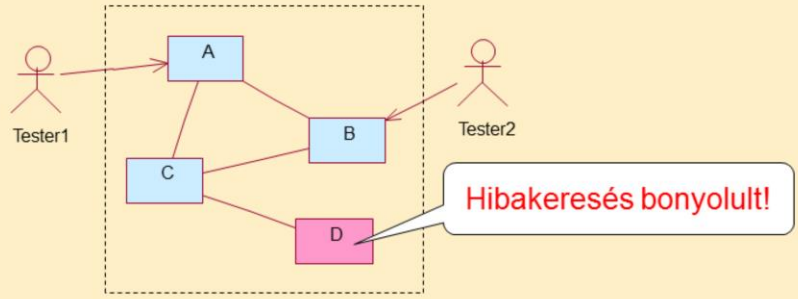
Modulok együttműködésének ellenőrzése

- Kapcsolódási interfészek tesztelése
  - A rendszer annak ellenére hibás lehet, hogy minden modul egyenként hibátlan!
- Módszerek:
  - Funkcionális tesztelés: Forgatókönyvek tesztje
    - Ez sokszor a specifikáció része (scenario)
  - (Strukturális tesztelés csak modulszinten!)
- Megközelítés:
  - "Big bang": minden modult egyszerre integrálni
  - Inkrementális: egyenként összerakni a modulokat

3

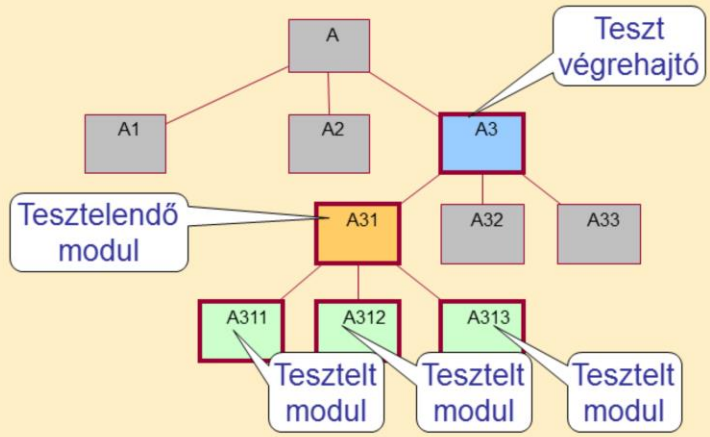
“Big bang” integrációs tesztelés

- Tesztelés a külső interfészeken keresztül
- Külső teszt végrehajtó
- Rendszer funkcionális specifikáció alapján történik
- Modul módosítás: Újratesztelés
- Kis rendszerek esetén alkalmazható



Alulról felfelé történő integrációs tesztelés

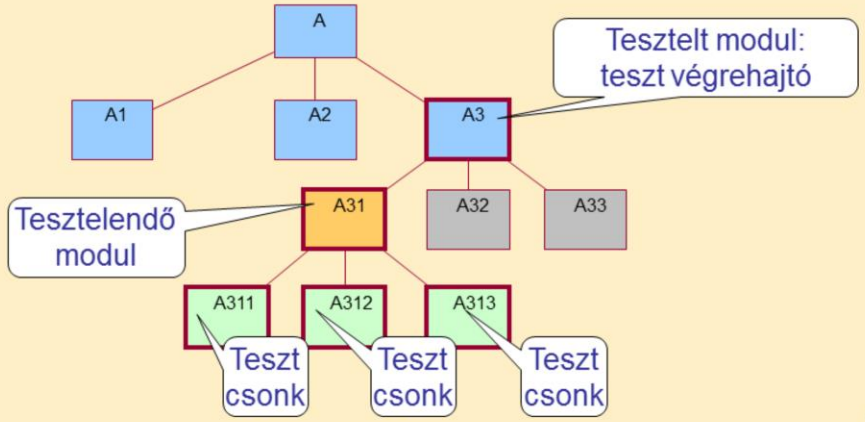
- Tesztelendő modul a már teszteltet használja
- Teszt végrehajtó szükséges
- Integrációval párhuzamosan megtehető
- Modul módosítás: felette lévők tesztjére hatással van



5

Felülről lefelé történő integrációs tesztelés

- Modulok a hívó modulokból kerülnek tesztelésre
- Csonkok helyettesítése tesztelendő modulokkal
- Erősen követelmény-orientált (“fentről” tesztelünk)
- Modul módosítás: alatta lévők tesztelését módosítja



Felülről lefelé vs. alulról felfelé

- Felülről lefelé
  - + Követelmény-orientált, ellenőrzéshez jól illeszkedő
  - + Hamar összeállhat egy demonstrálható „szkeleton”
  - - Csonkok készítése általában nehezebb
  - - Teszt bemenetek távol lehetnek az épp integrálandó modultól
- Alulról felfelé
  - + Integráció orientált, konstruktívabb
  - + Könnyebb megfigyelni és irányítani a teszteket
  - - A rendszer maga csak a legvégén áll össze

7

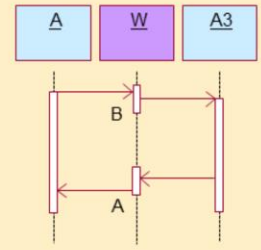
Futtató rendszer integrációja

- Motiváció:
  Nehéz csonkokat írni a futtató rendszerhez
  - Pl. OS, J2EE konténer, …
- Stratégia:
  1. Alkalmazás modulok integrációja felülről lefelé, a futtató rendszer szintjéig
  2. Futtató rendszer alulról felfelé történő tesztelése
     - Funkciók izolációs tesztje (ha szükséges)
     - „Big bang" tesztelés az alkalmazás hierarchia legalsó rétegével
  3. Alkalmazás és futtatórendszer integrációja, a felülről lefelé történő tesztelés befejezése

8

Eszközök az integrációs teszteléshez

- Wrapper (csomagoló) kódrészletek
  - „before” wrapper: A hívás végrehajtása előtt
    - Paraméterek vizsgálata
    - Hívási szekvencia ellenőrzése
  - „after” wrapper: A hívás végrehajtása után
    - Visszaadott érték ellenőrzése vagy módosítása
  - „replace” wrapper: A hívott helyettesítése
    - Teszt csonk megvalósítás



9

Tartalom

- Modul / Unit tesztelés
- Integrációs tesztelés
- Rendszer tesztelés
- Elfogadás tesztelés

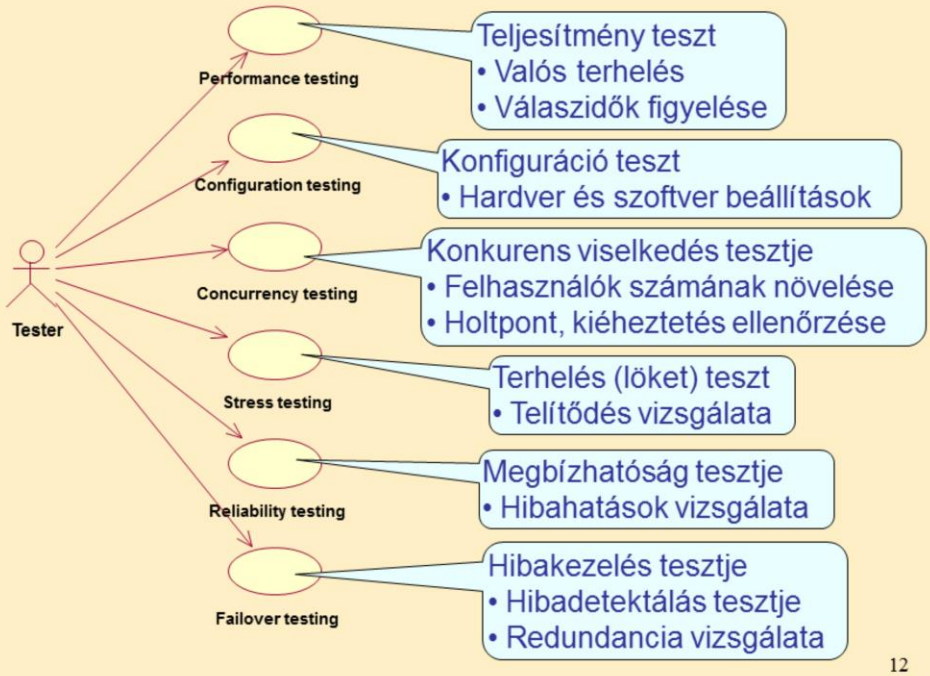
- Tesztek leírása: U2TP

Rendszertesztelés

Tesztelés a rendszerszintű specifikáció alapján

- Jellemzők:
  - Hardver-szoftver integráció után végezhető
  - Funkcionális tesztek +
    nem-funkcionális jellemzők tesztje is!
- Kiemelhető:
  - Adat integritás vizsgálata
  - Felhasználói profil figyelembe vétele (terhelés)
  - Rendszer alkalmazhatósági korlátok megállapítása (erőforrás-használat, telítődés)
  - Hibahatások vizsgálata

Rendszerteszt típusok



12

Tartalom

- Modul / Unit tesztelés
- Integrációs tesztelés
- Rendszer tesztelés
- Elfogadás tesztelés

- Tesztek leírása: U2TP

Elfogadás tesztelése (validációs tesztelés)

- Cél: Valóságos környezet hatásának tesztelése
  - Felhasználói elvárások figyelembe vétele:
    Nem specifikált felhasználói elvárások is megjelennek
  - Váratlan eseményekre való reagálás:
    Kis valószínűségű eseménykombinációk is megjelennek
- Időzítési szempontok
  - Időzítések megfigyelése az adott környezetben
  - Korlátok ellenőrzése: Valósidejű monitorozás
- Környezeti szimuláció
  - Adott szituációk valóságban nem tesztelhetők
    (pl. védelmi rendszerek)
  - Szimulátorokat is validálni kell

Összefoglalás: Tesztelési feladatok

1. Modul/unit tesztelés
   - Izolációs tesztelés
2. Integrációs tesztelés
   - „Big bang” tesztelés
   - Top-down (felülről-lefelé) tesztelés
   - Bottom-up (alulról-felfelé) tesztelés
   - Futtató környezet integrációja
3. Rendszertesztelés
   - Teljes rendszer együttes tesztelése
4. Validációs tesztelés
   - Felhasználói elvárások tesztelése
   - Környezeti szimuláció

15

Összefoglalás: Különbség a szintek között

How Google Tests Software könyv ajánlásai:

	Small	Medium	Large
Javasolt futási idő	< 100 ms	< 1 sec	minél gyorsabban
Időkorlát (kill)	1 perc	5 perc	1 óra

Erőforrás	Small	Medium	Large
Hálózat (socket)	Mocked	csak localhost	Igen
Adatbázis	Mocked	Igen	Igen
Fájl hozzáférés	Mocked	Igen	Igen
Rendszerhívás	Nem	Nem javasolt	Igen
Több szál	Nem javasolt	Igen	Igen
Sleep utasítás	Nem	Igen	Igen
Rendszer tulajdonságai	Nem	Igen	Igen

16

Tartalom

- Modul / Unit tesztelés
- Integrációs tesztelés
- Rendszer tesztelés
- Elfogadás tesztelés

- Tesztek leírása: U2TP

U2TP: UML 2 Testing Profile (OMG, 2004)

- Able to capture all needed information for functional black-box testing (specification of test artifacts)
  - Mapping rules to TTCN-3, JUnit
- Language (notation) and not a method (how to test)

Packages (concept groups):

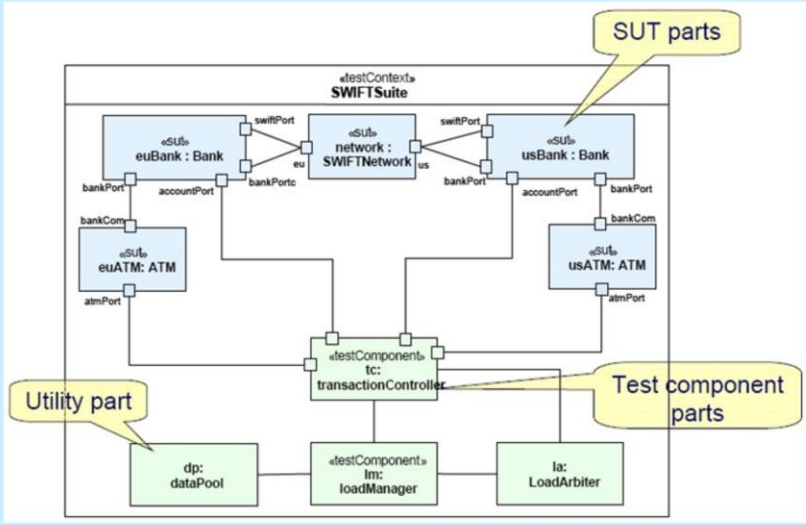
- Test Architecture
  - Elements and relationship involved in test
  - Importing the UML design model of the SUT
- Test Data
  - Structures and values to be processed in a test
- Test Behavior
  - Observations and activities during testing
- Time Concepts
  - Timer (start, stop, read, timeout), TimeZone (synchronized)

U2TP Test Architecture package

Identification of main components:

- SUT: System Under Test
  - Characterized by interfaces to control and observation
  - System, subsystem, component, class, object
- Test Component: part of the test system (e.g., simulator)
  - Realizes the behavior of a test case
    (Test Stimulus, Test Observation, Validation Action, Log Action)
- Test Context: collaboration of test architecture elements
  - Initial test configuration (test components)
  - Test control (decision on execution, e.g., if a test fails)
- Scheduler: controls the execution of test components
  - Creation and destruction of test components
- Arbiter: calculation of final test results
  - E.g., threshold on the basis of test component verdicts

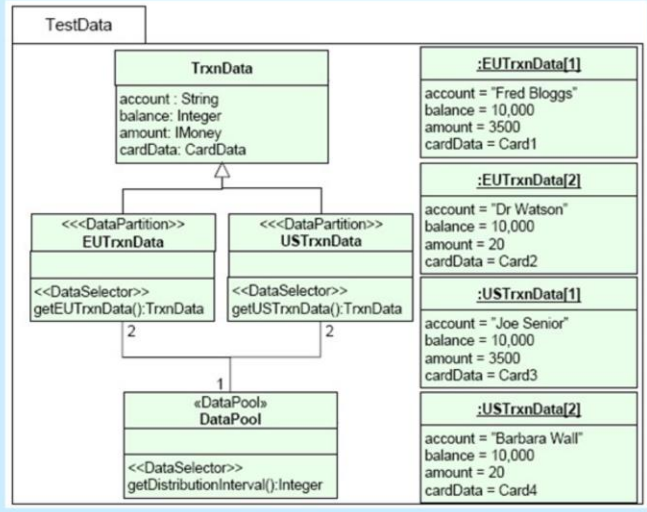
U2TP Test Architecture example



U2TP Test Data package

- Identification of types and values for test (sent and received data)
  - Wildcards (* or ?)
  - Test Parameter
    - Stimulus and observation
  - Argument
    - Concrete physical value
  - Data Partition: Equivalence class for a given type
    - Class of physical values, e.g., valid names
  - Data Selector: Retrieving data out of a data pool
    - Operating on contained values or value sets
  - Templates

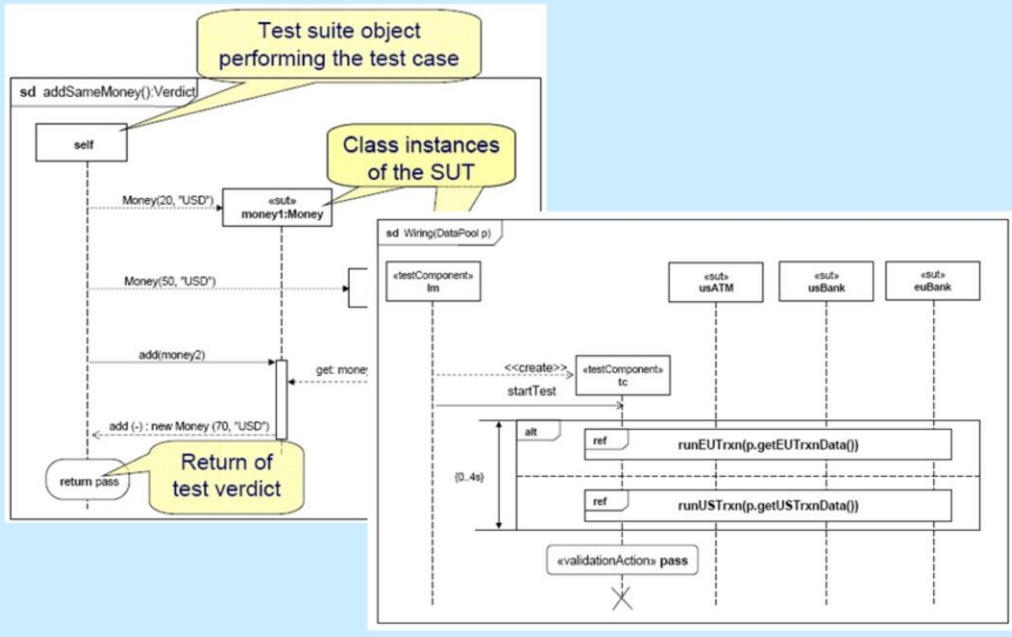
U2TP Test Data example



U2TP Test Behavior package

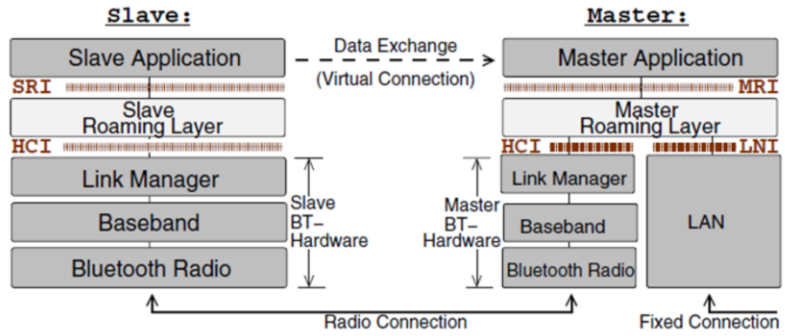
- Specification of default/expected behavior
- Identification of behavioral elements:
  - Test Stimulus: test data sent to SUT
  - Test Observation: reactions from the SUT
  - Verdict: pass, fail, error, inconclusive values
  - Actions: Validation Action (inform Arbiter), Log Action
- Test Case: Specifies one case to test the SUT
  - Test Objective: named element
  - Test Trace: result of test execution
    - Messages exchanged
  - Verdict

U2TP Test Behavior example



Mintapélda: BlueTooth roaming

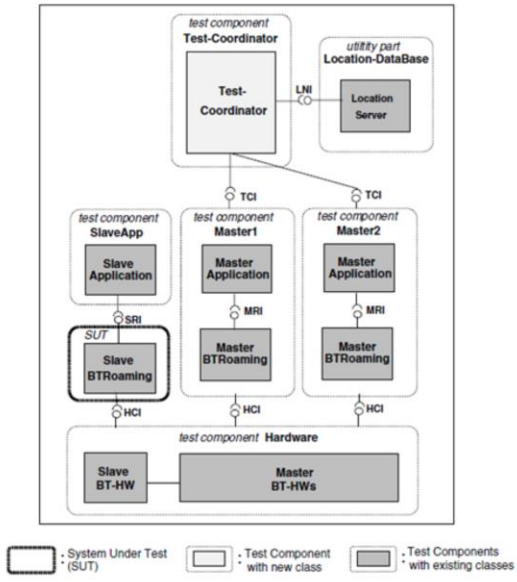
Tesztelendő rendszer:



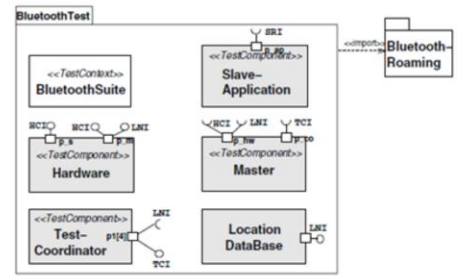
Teszt cél:

- Slave Roaming Layer funkciói
  - Link minőségének figyelése
  - Kapcsolat létesítése másik masterrel

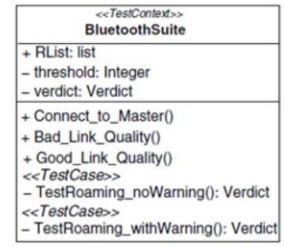
Komponensek szerepei



Overview

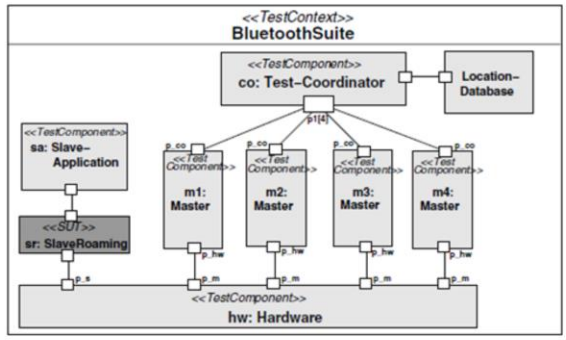


Test package

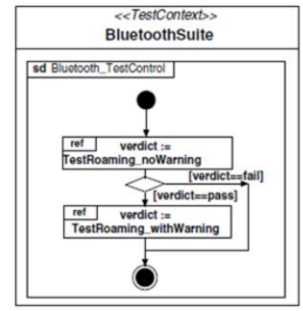


Test context

Teszt konfiguráció és teszt vezérlés



Test configuration

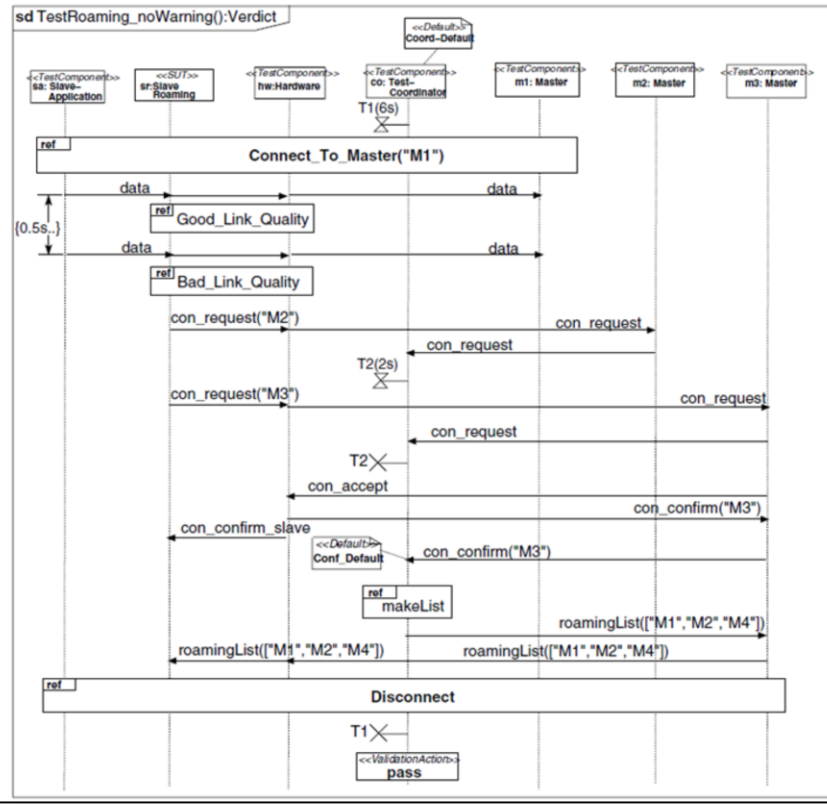


Test control

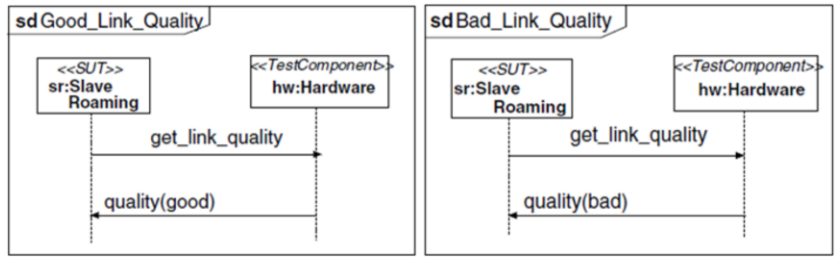
Test scenario

Test case implementation (see BlueToothSuite)

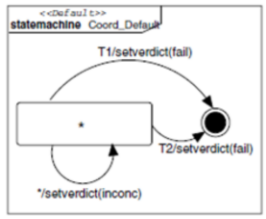
- References
- Timers
- Defaults



Néhány hivatkozott részlet



Sequence diagrams



Default behaviours specified to catch the observations that lead to verdicts

- Here: Processing timer events