Acceptance Test Optimization

Mohamed Mussa, Ferhat Khendek

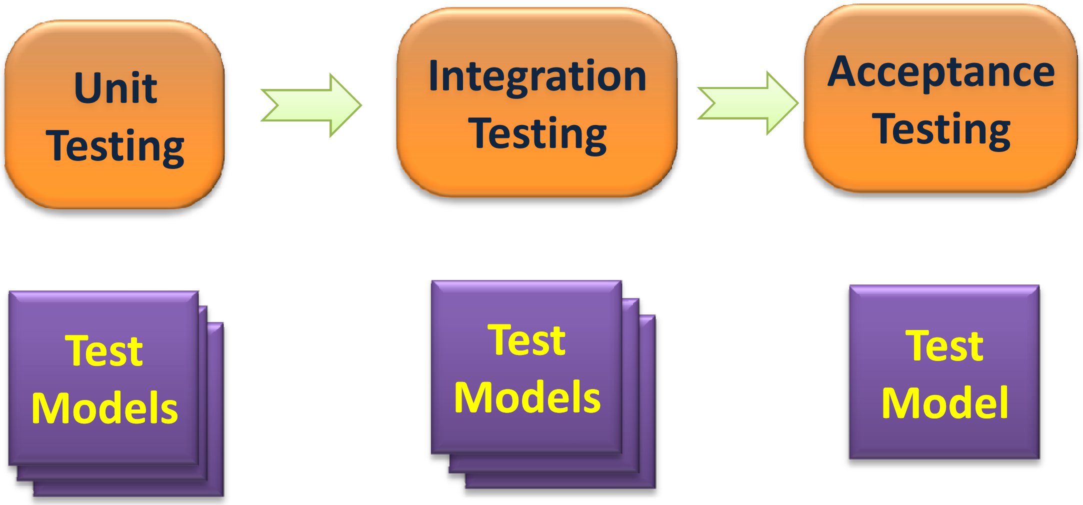
SAM'2014

Outline

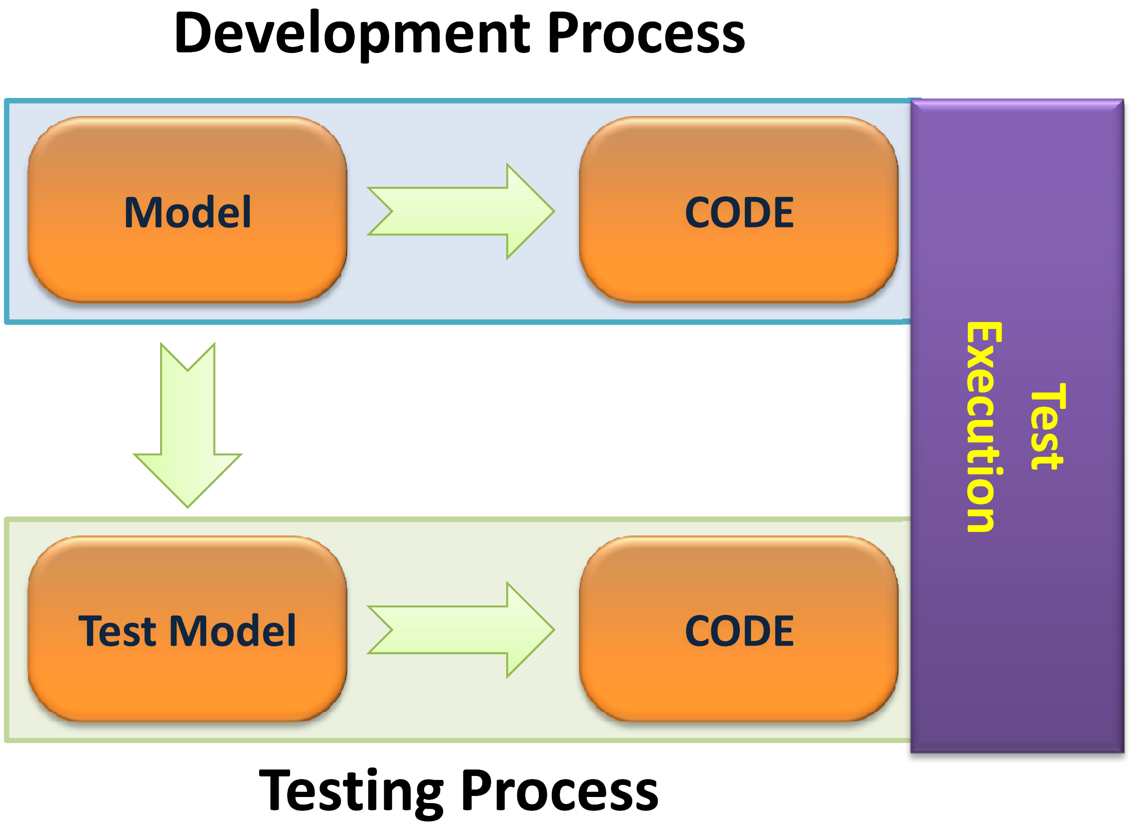
- Background
- Problem Statement
- Overall Approach
  - Integration test cases selection
  - Comparing test models
- Conclusion

Background

Test process consists of several phases



Background - Model Based Testing



Development Process

Model

CODE

Test Model

CODE

Testing Process

Test Execution

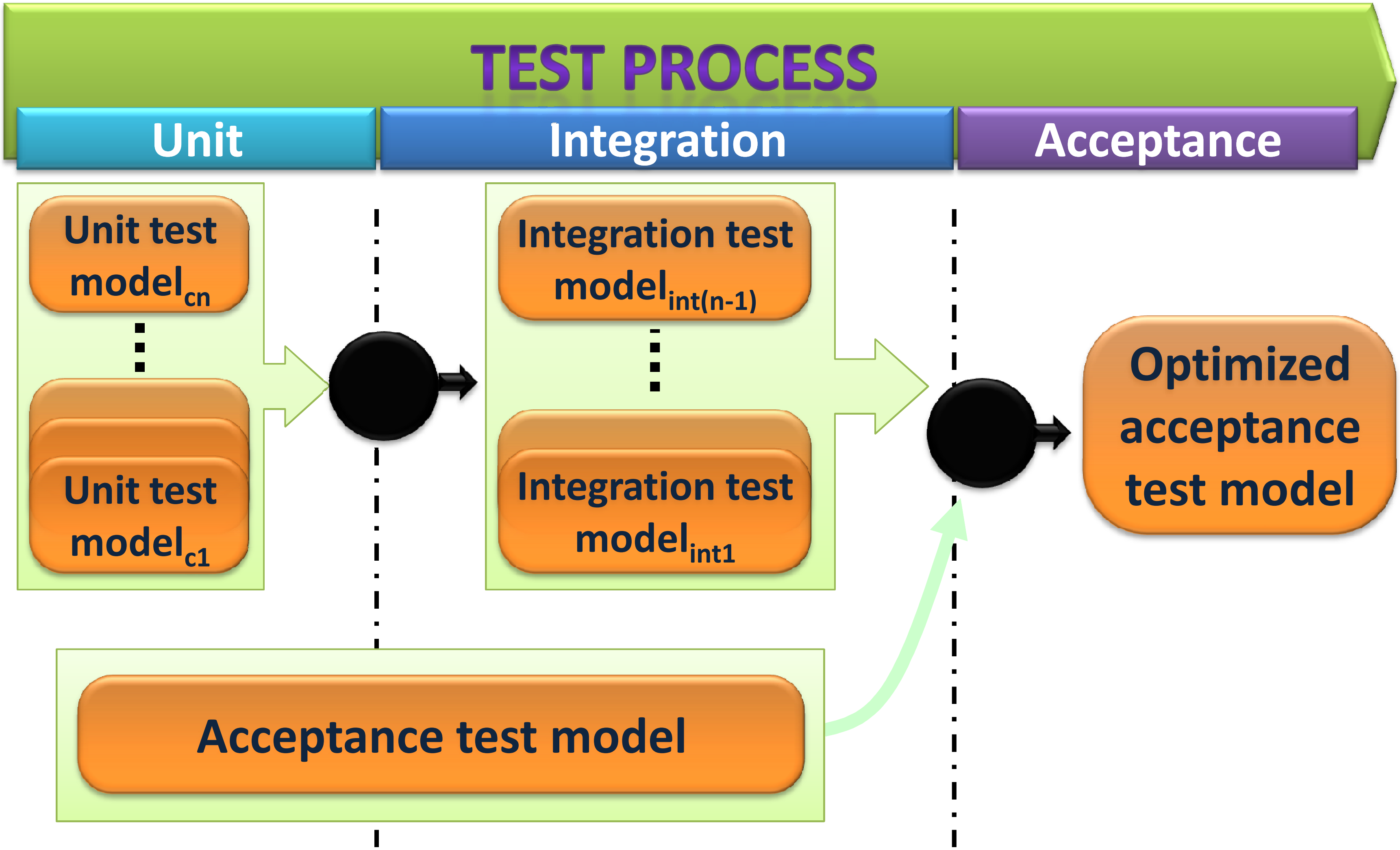
Background - Model Based Testing

- Different approaches for different test phases
  - Unit, integration, acceptance
- Different notations/languages
  - Different subsets of the same language
- Many test models are produced during different phases
  - Redundancy
  - **Test generation/planning: no reusability**
  - **Test Execution: no optimization**

Background - Model Based Test Framework

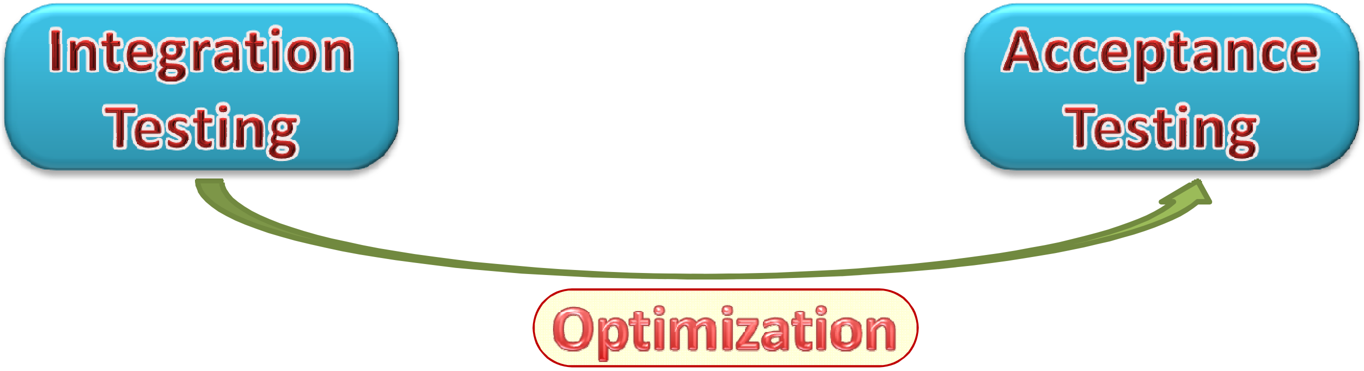
- Goals
  - Provide a systematic transition between the test phases
    - Framework
  - Strengthen the collaboration between the development and the testing teams
  - Well know standards & reuse
  - Improve the test process
    - Enable reusability & optimization

Background - Model Based Test Framework



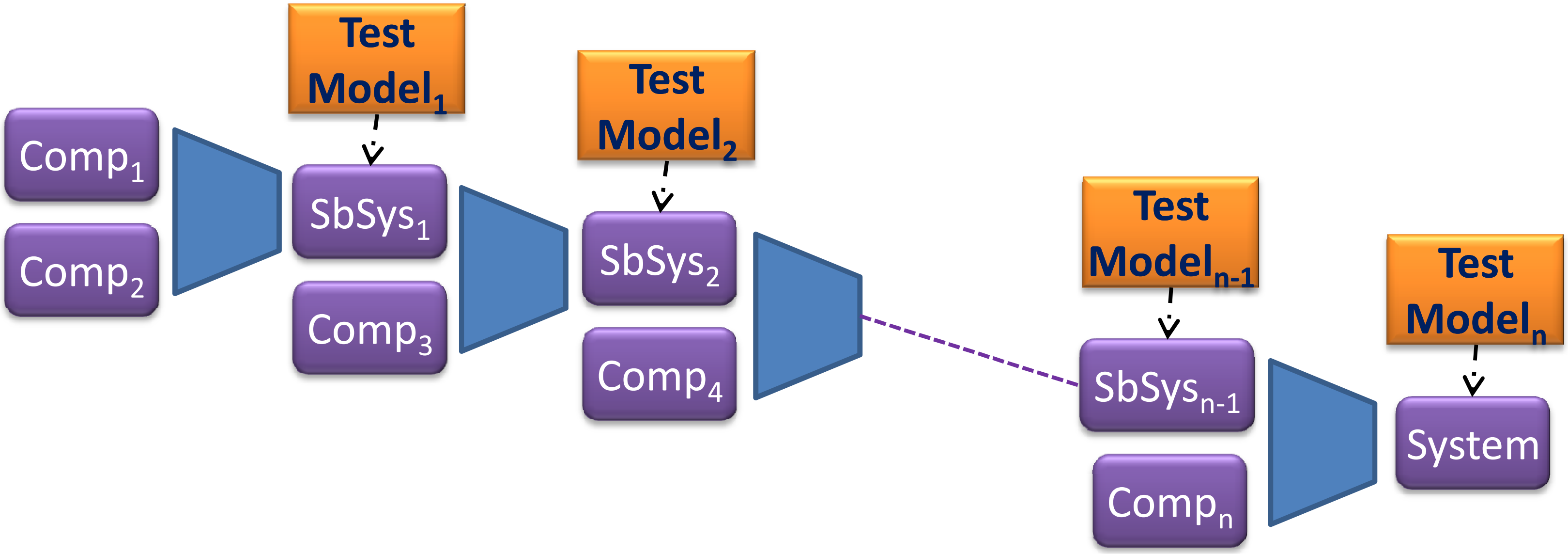
Problem Statement

- Test cases may be exercised several times across the testing phases
  - Integration vs. Acceptance
- **Goal: remove redundant acceptance test cases**
  - Reduce test execution time



Problem Statement

- Obvious solution
  - Compare integration test cases and acceptance test cases
- Problem
  - **Some integration test cases may include stubs for subsequent system components**
  - Cannot be substituted to acceptance test cases

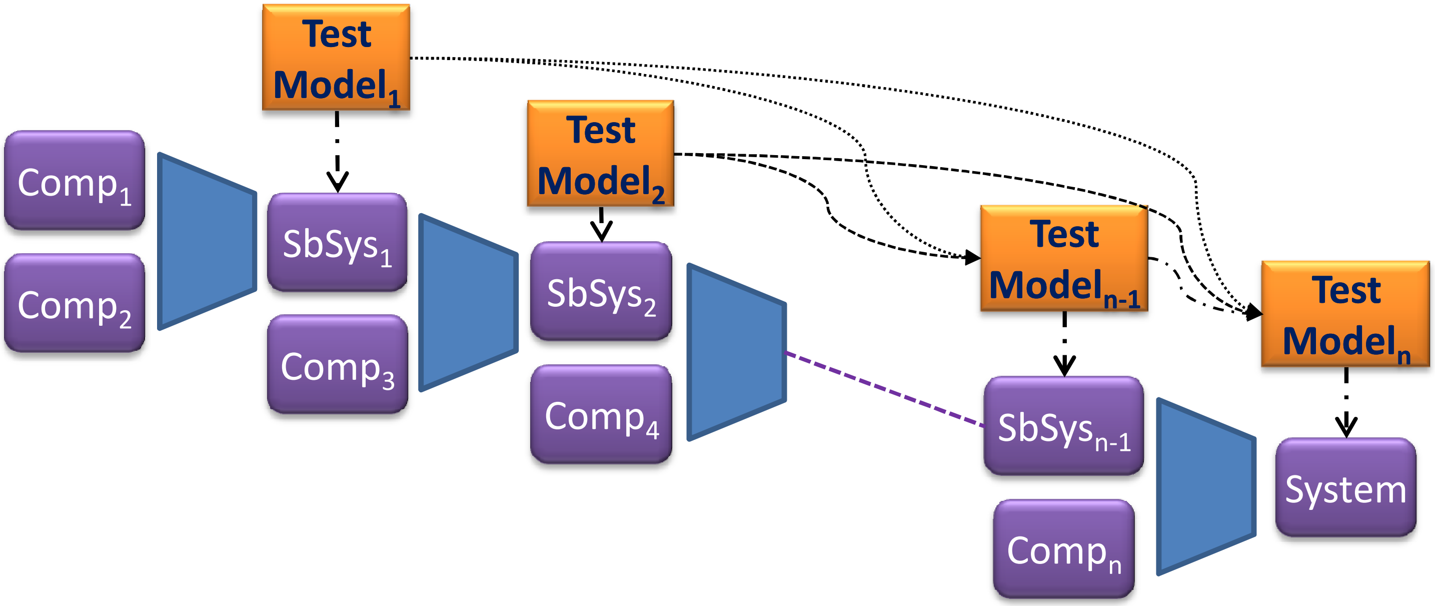


Overall Approach

- Integration test cases selection
- Compare integration test cases to acceptance test cases

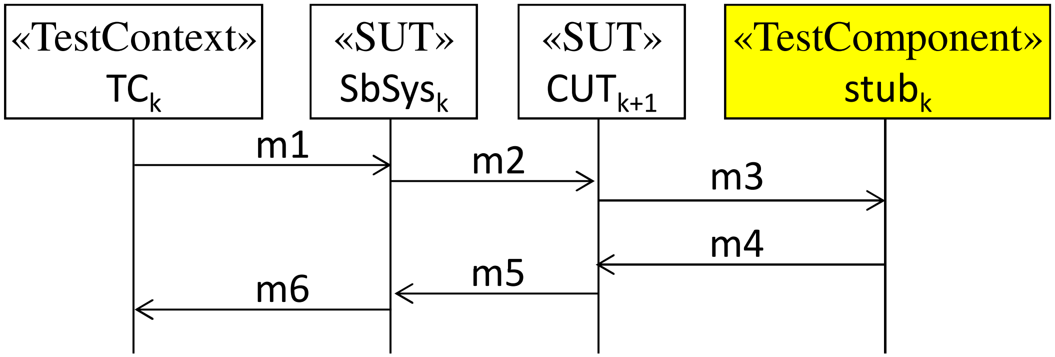
Integration test cases selection

- Test cases of last integration round are applied on complete system
- Compare the behavior of test stubs of each test case to the behavior of CUTs of test cases of subsequent integration rounds
- No additional information beside the test models

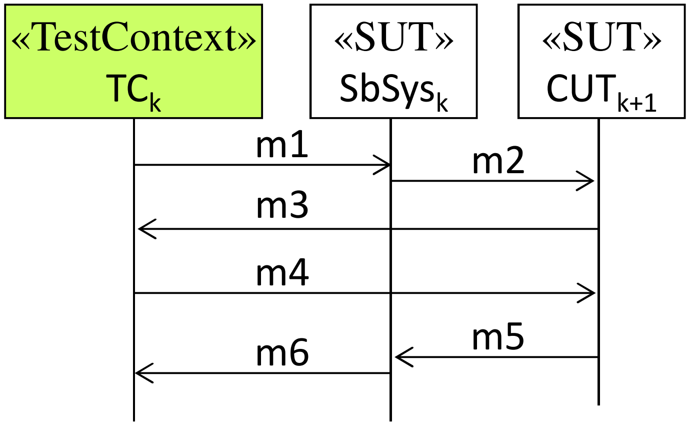


Integration test cases selection

Test stubs can be specified explicitly, or



specified implicitly



Integration test cases selection

- Event based comparison
- Not instance based comparison
  - Instances are different

- Not event name based
  - but message
  - event types: message, time, miscellaneous

Integration test cases selection

Selection condition

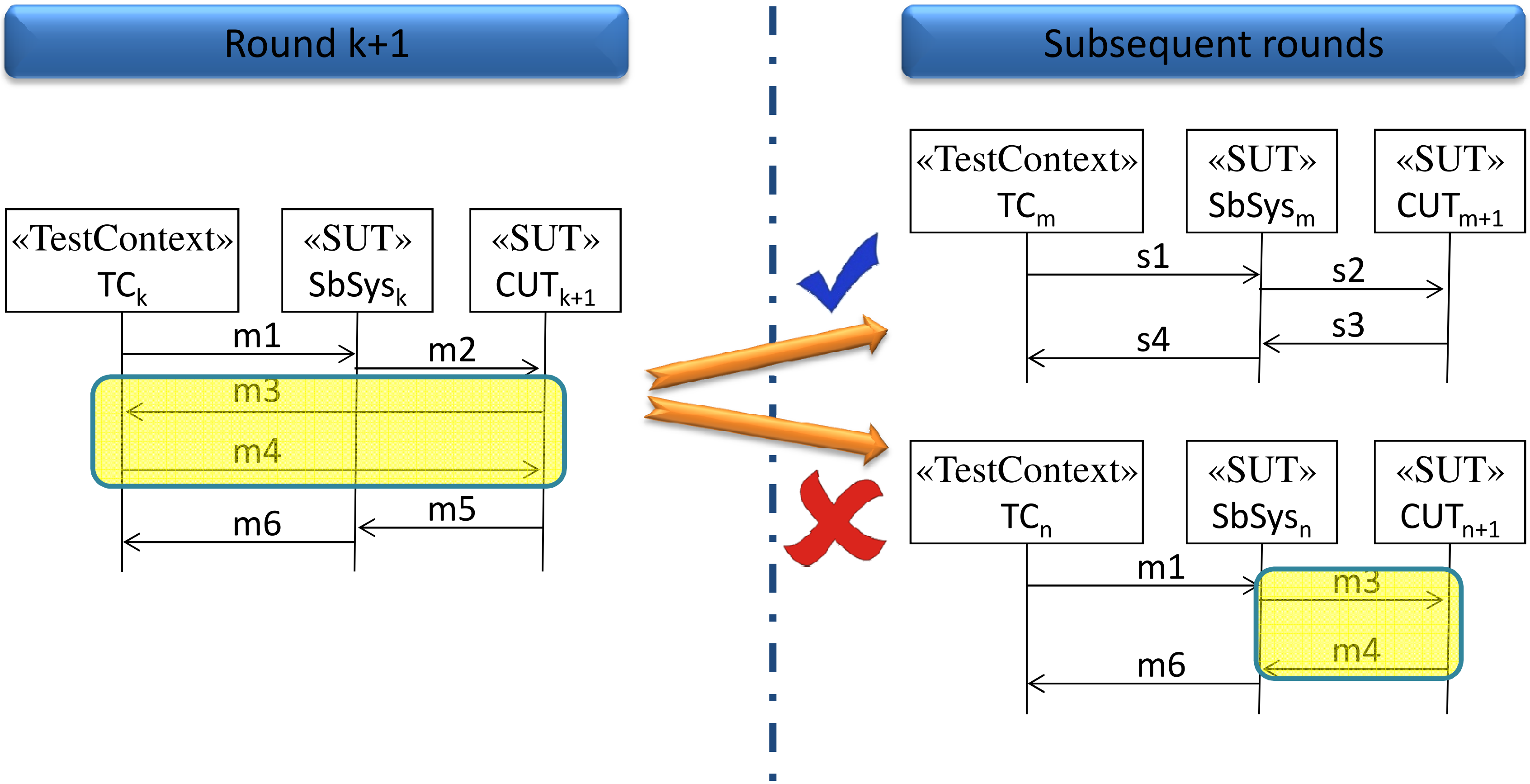
- Let
  - $T_k = \{I_k, E_k, R_k\}$ be an integration test case at integration round ***k*,**
  - $T_i = \{I_i, E_i, R_i\}$ be an integration test case at integration round ***i*,**
  - $i > k$

- *T_k does not use a test stub for the CUT of T_i if and only if*

$$\forall(e_i, e_k) \cdot e_i \in E_i, e_k \in E_k \mid (e_i \neq e_k) \vee ((e_i = e_k) \wedge (e_i.owner.st \neq SUT)).$$

Integration test cases selection

Comparing integration test cases



Integration test cases selection

Test cases, which their stubs do not match with subsequent CUTs, are compared to acceptance test cases

Comparing Test Models

- A lot of work has been done
  - Compared models are evolved from the same source
  - Two-Way vs. Three-Way
  - Look up for differences (Add/Delete/Modify)
  - Structure vs. Behavior

- Our case
  - Models did not necessary evolve from the same source

Comparing Test Models

- Comparing MSCs or Sequence Diagrams is not straightforward
  - *Several researchers have tackled this issue*

- *But this is not difficult for test cases*
  - *Finite behaviors*

Comparing Test Models

- A test case *T is a tuple (I, E, R), where*
  - *I : a set of instances*
  - *E : a set of events*
  - *R* $\subseteq$ *(E x E): a partial order reflecting the transitive closure of the order relation between events on the same axis and the sending and reception events of the same message*

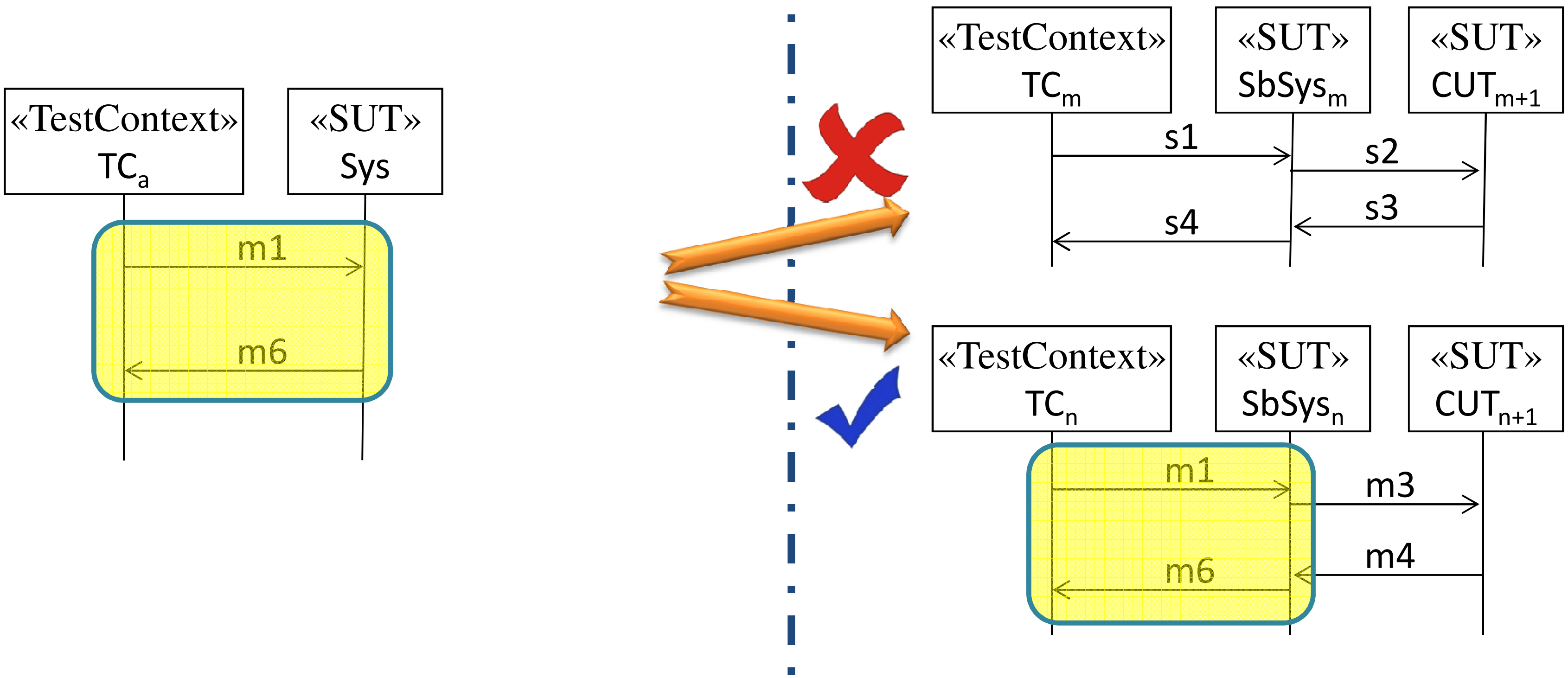
- **Test case inclusion**
  - $T_{acc} = \{I_a, E_a, R_a\}$ and $T_{int} = \{I_i, E_i, R_i\}$
  - T_{acc} *is included in* T_{int} iff
    - $E_a \subseteq E_i$
    - $R_a \subseteq R_i$

Comparing Test Models

Comparing test cases

Acceptance test case

Integration test cases



Conclusion

- We proposed an optimization approach that reduces the acceptance test suite length
  - already done at integration phase
- Implemented and completed the framework
- What kind of systems would benefit ?
- Requires evaluation of the gain