

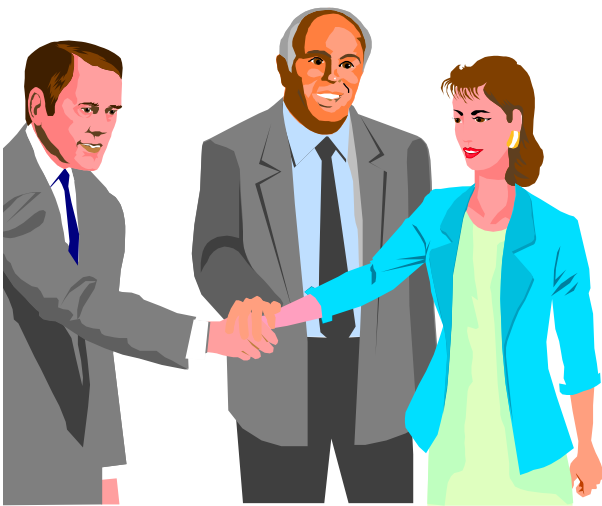
Automatic Synthesis of SDL models in Use Case Methodology

Dr. Nikolai N. Mansurov

Dr. Nikolai N. Mansurov, Dept. of CASE tools, ISP

Dr. Nikolai N. Mansurov

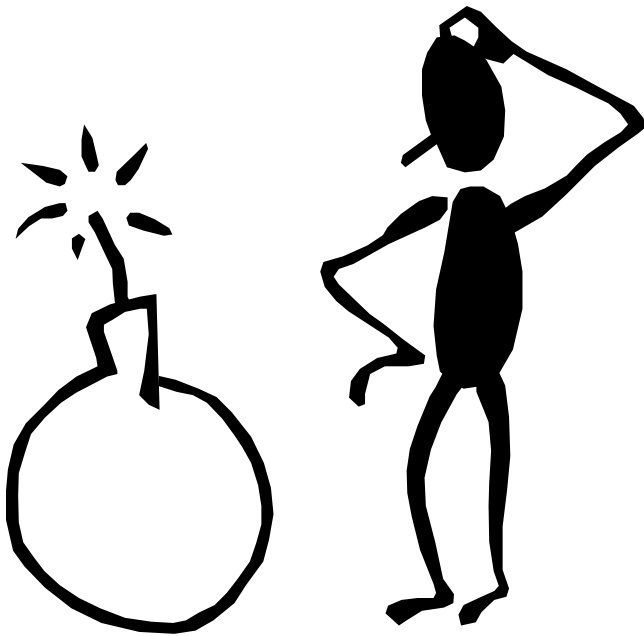
Head of Dept. of CASE Tools, Institute for System
Programming, Moscow



- Formal description techniques: SDL, MSC, UML, ASN.1, ACT-1
- Compiler Design
- Automatic Code Generation
- Program understanding, reverse engineering

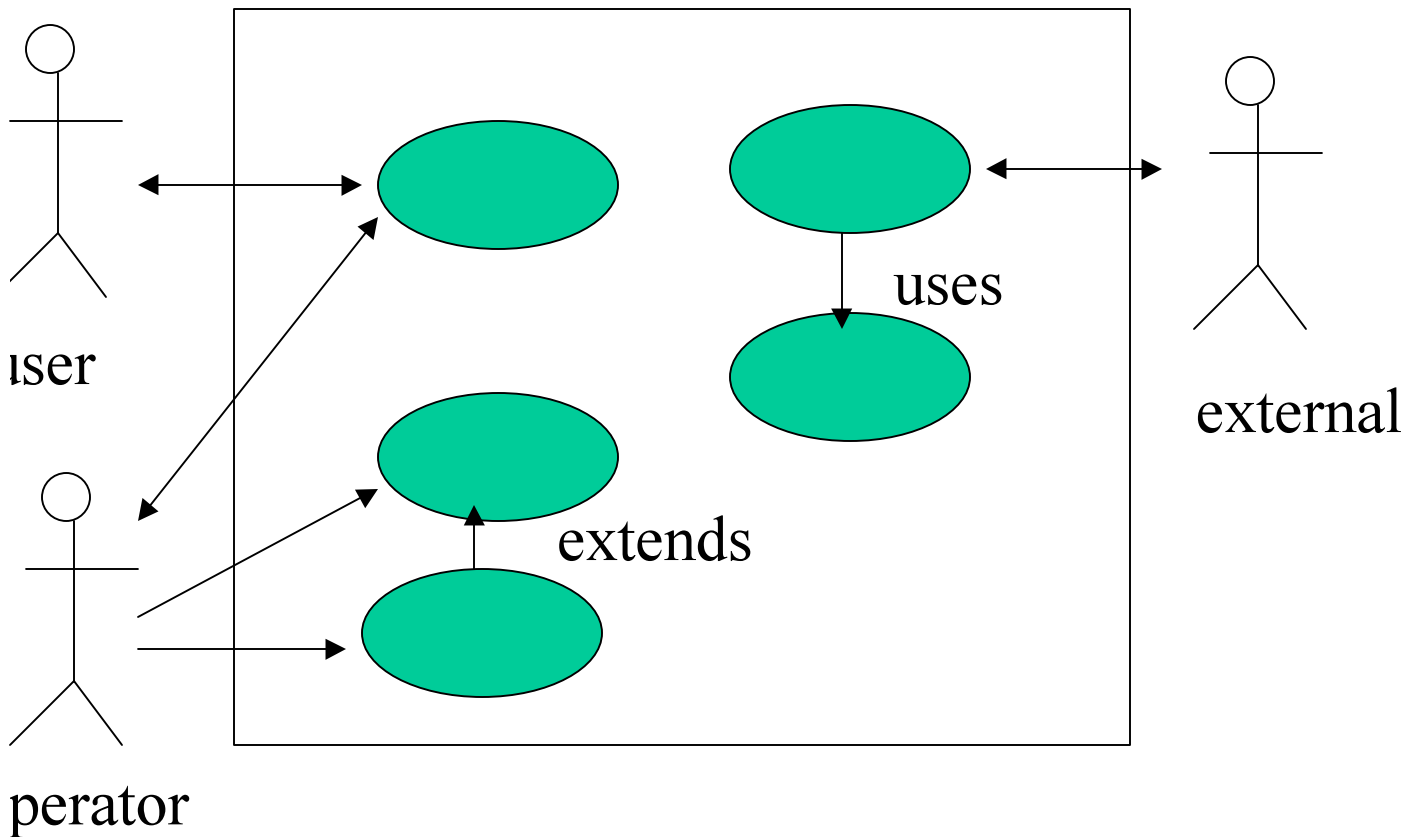
Dr. Nikolai N. Mansurov, Dept. of CASE tools, ISP

Definition of Use Case



- Actor
 - external entity, communicates with system to achieve goal
- Use Case
 - sequence of interactions between one or more actors and system
- Scenario
 - sequence of events

Description of Use Cases



Description of Use Cases

Use Case diagram

- relations between actors and use cases
- relations between use cases

Description of scenarios

- informal text
- structured text/tabular
- UML Sequence Diagram
- Message Sequence Charts (MSC)

Overview of the Process

Analysis

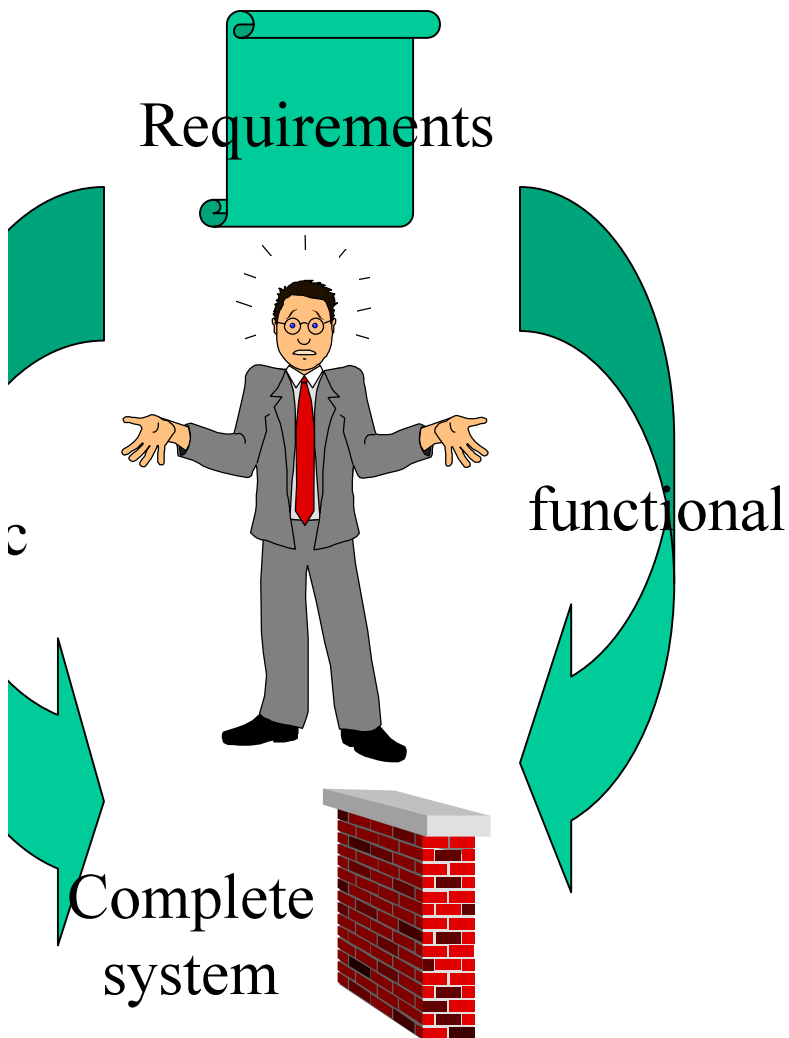
- Requirements Analysis
- System Analysis

Design

- System Design
- Detailed Design

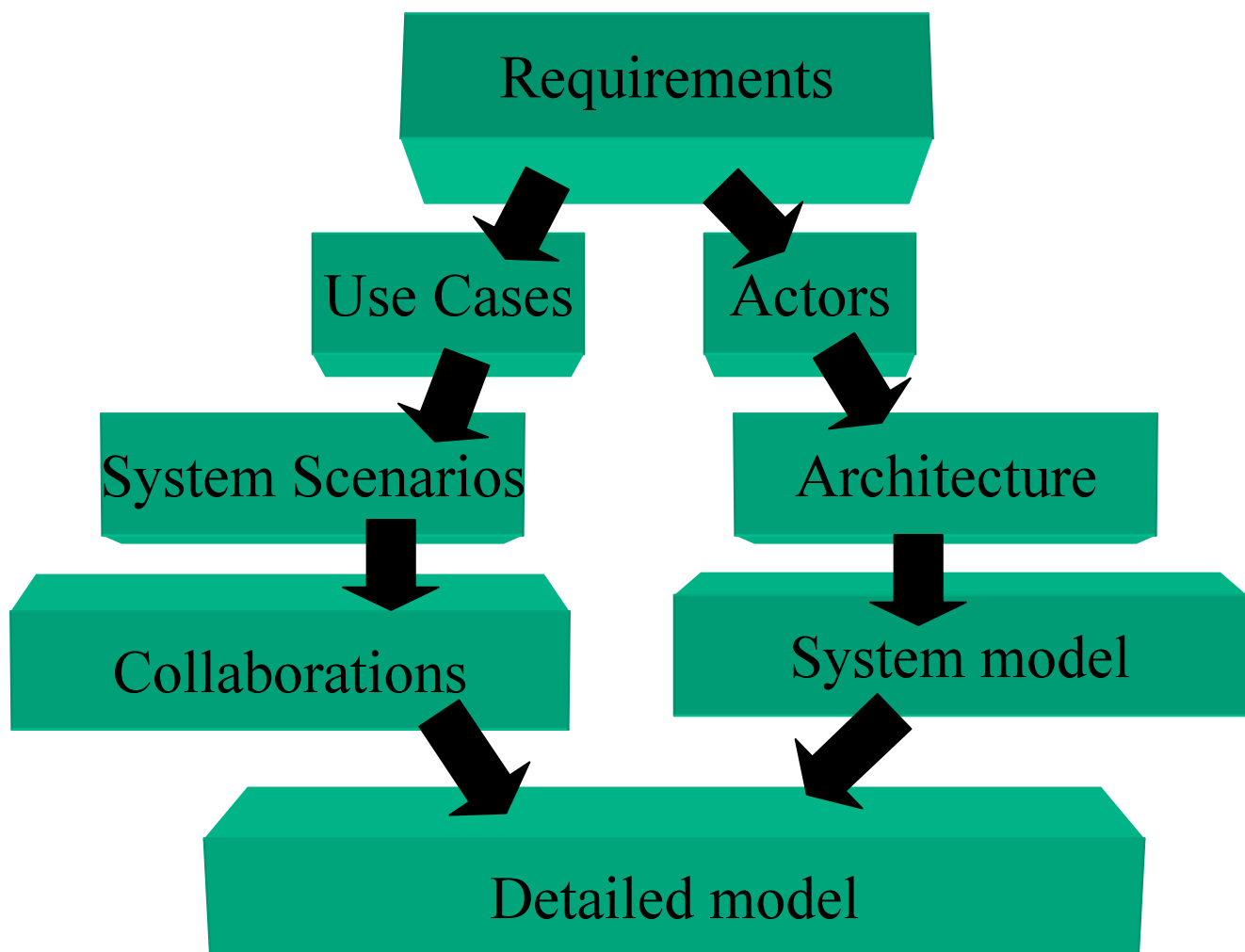
Implementation

Two Modeling Perspectives

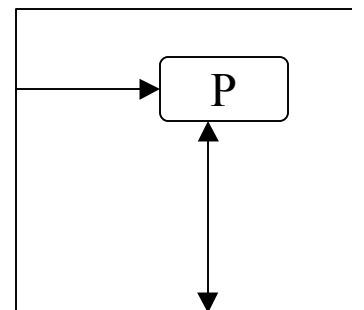
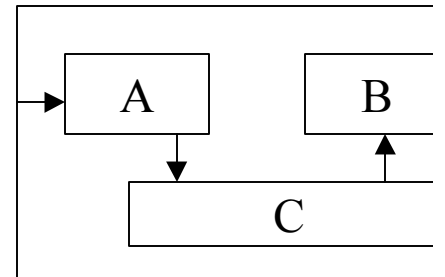
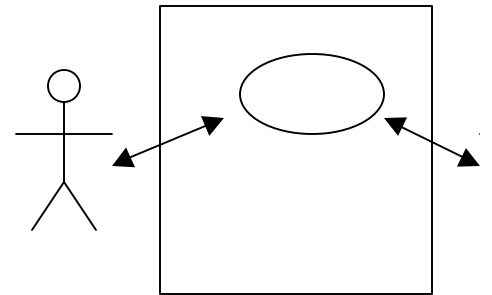
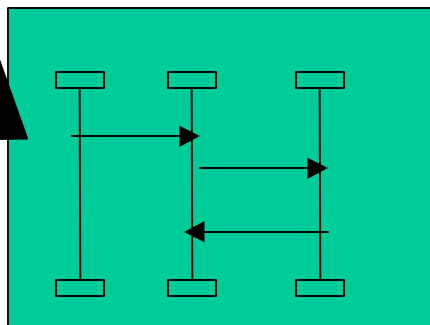
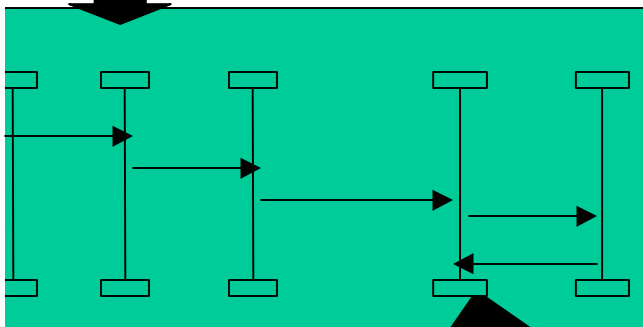
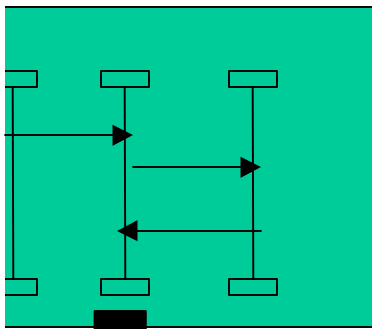


- Static Path
 - how entities form structures and hierarchies
- Functional Path
 - how entities collaborate

Use Case Methodology



Behavior and Structure



Summary of Approach

Formalization of Use Cases using HMSC

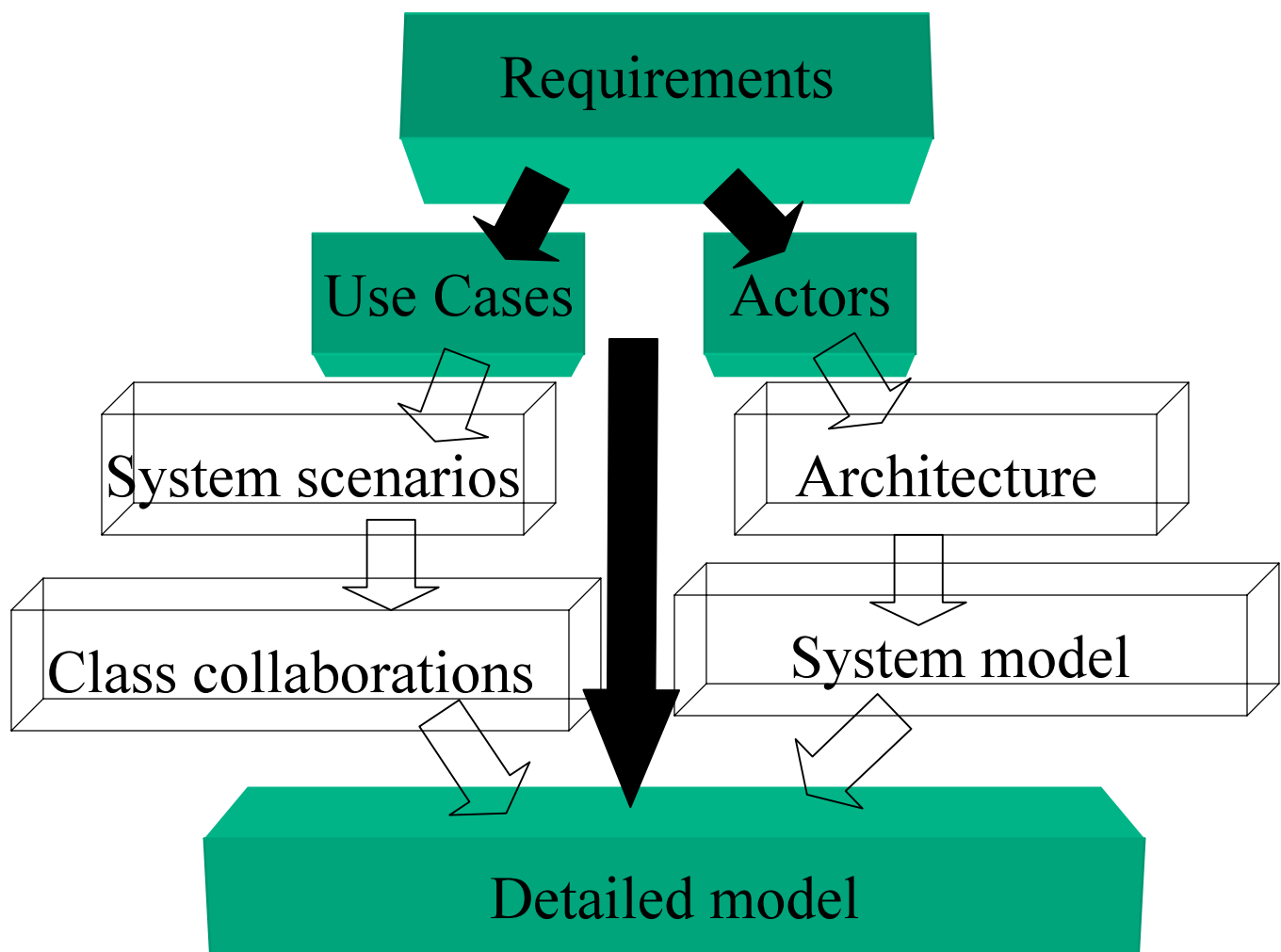
Use Cases are mapped to FSM

Use Cases describe Data Flow

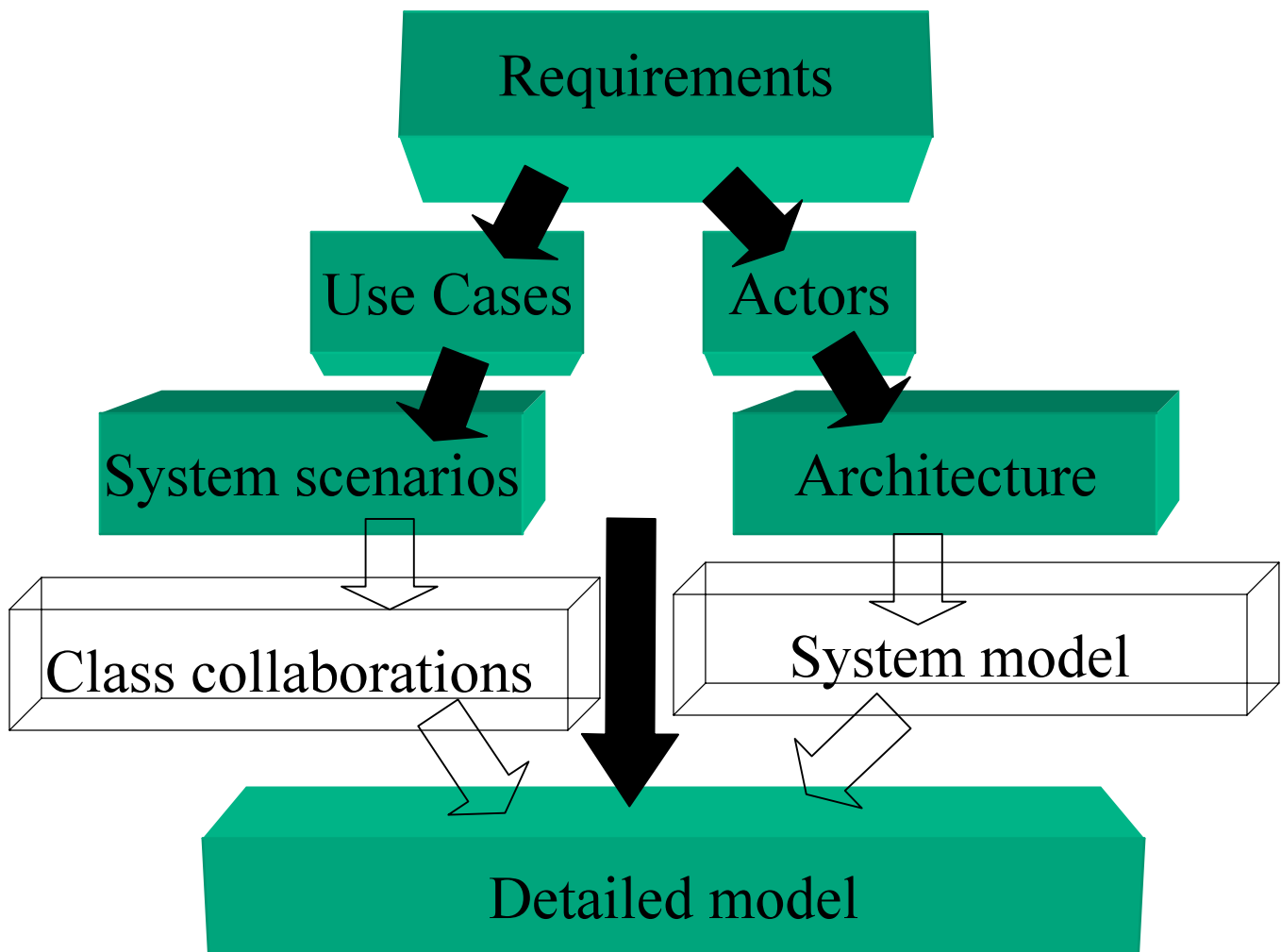
Automatic Synthesis of SDL models

- complete
- both static structure & behavior
- typebased
- non-deterministic

Summary of Approach (2)



Summary of Approach (3)



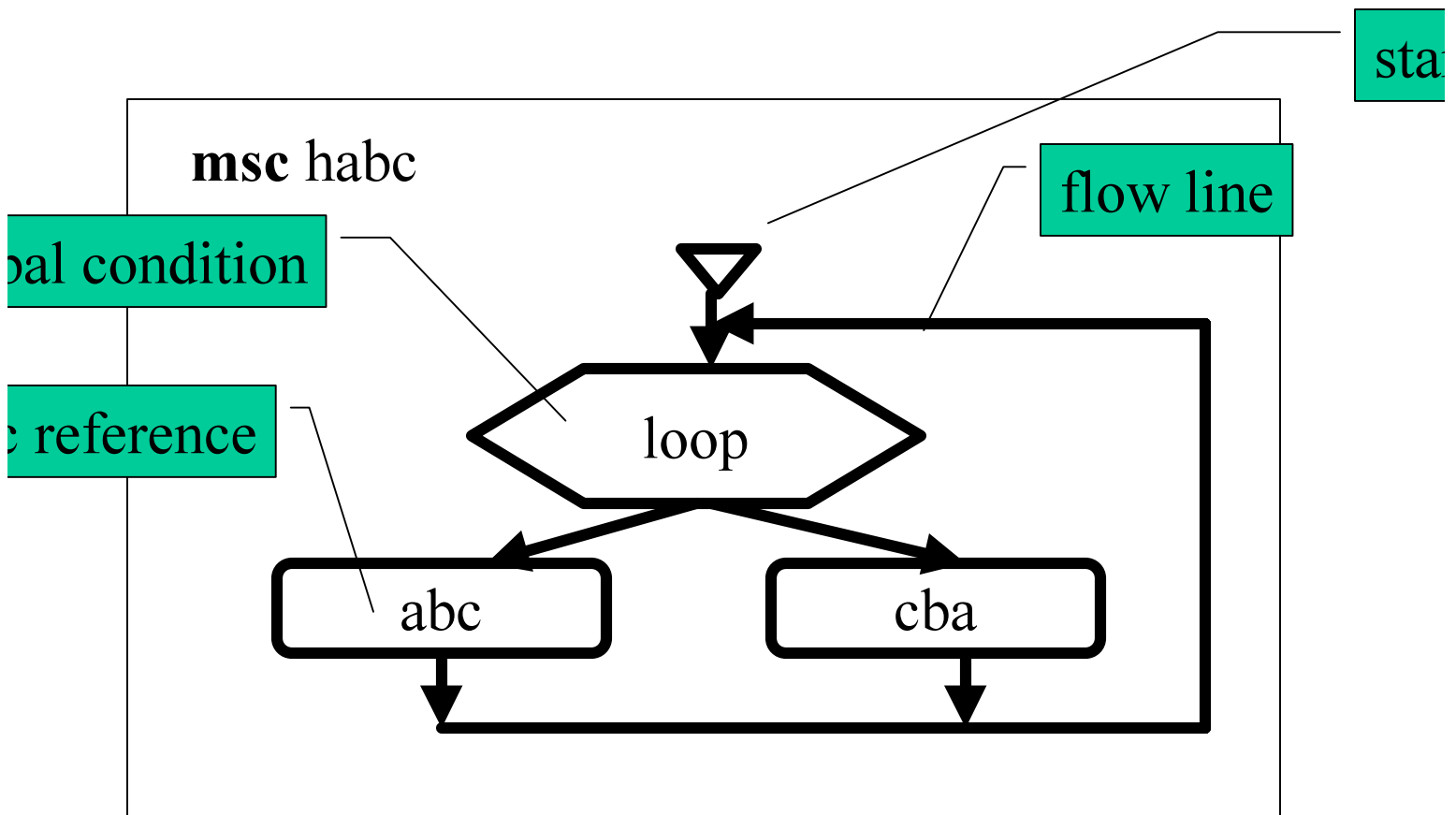
Formalization of Use Cases

Each scenario is formalized using *Message Sequence Charts* (MSC)

Control-flow relationships between scenarios are formalized using *High-Level MSC* (HMSC)

Data-flow relationships between scenarios are formalized using our *Data Extensions* to MSC language

High-Level MSC (HMSC)



Control-flow relationships

Alternative (sub-) scenarios

Iterations of (sub-) scenarios

“Uses” relation between use cases

“Extends” relation between use cases

Execution of use cases

Sequential schema

- use cases can not execute simultaneously

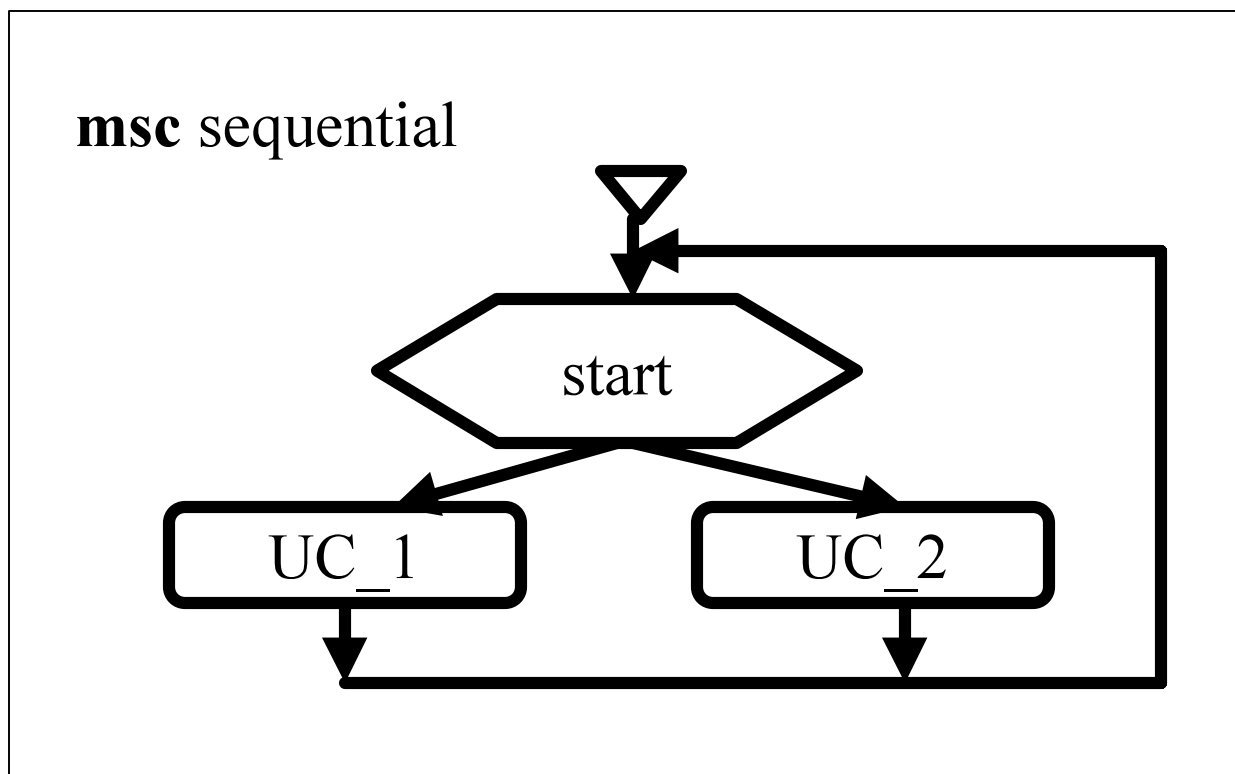
Parallel schema

- different use cases can execute simultaneously

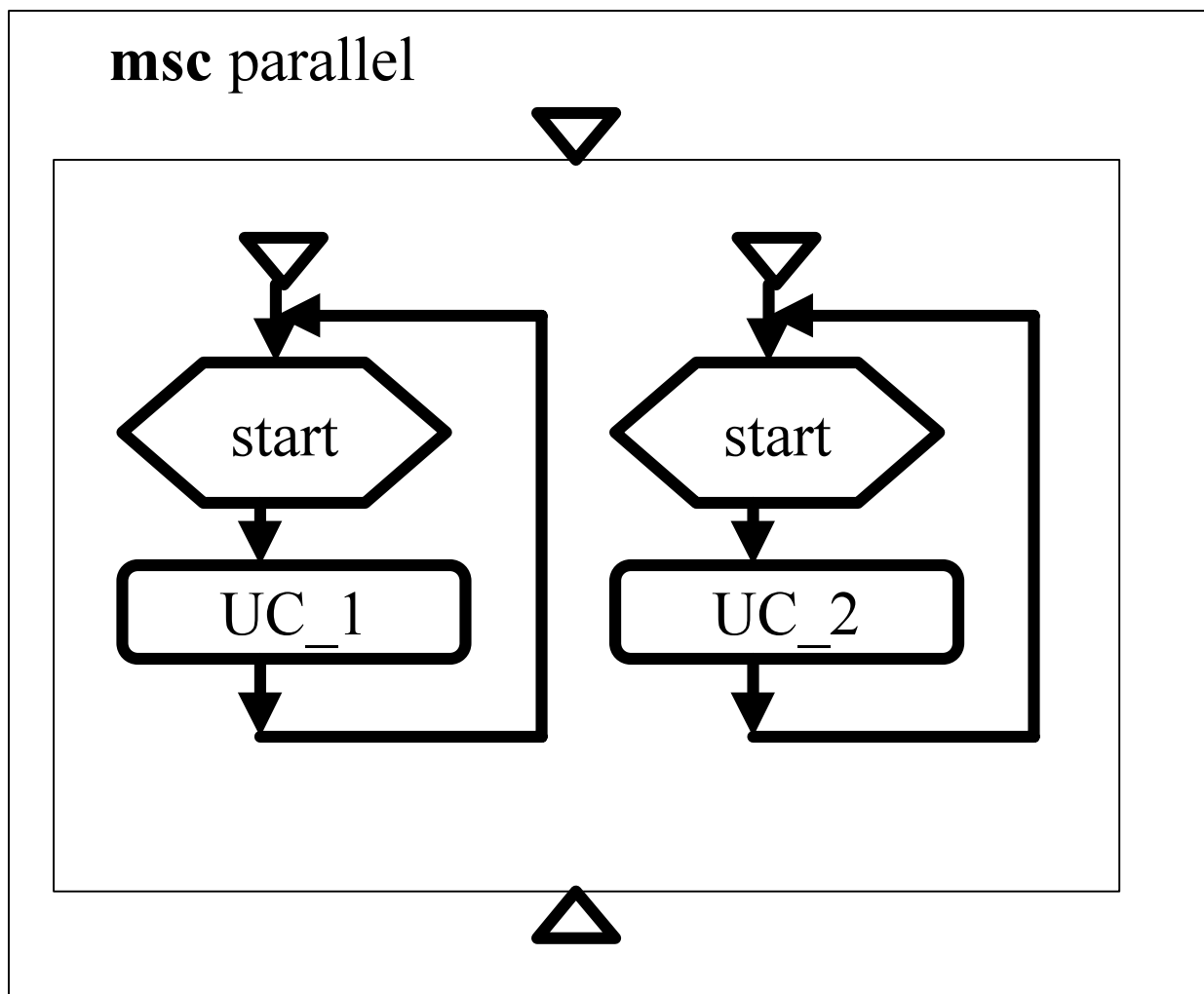
Multiple instance schema

- multiple instances of the same use case can execute simultaneously

Sequential schema



Parallel schema



Data Flow

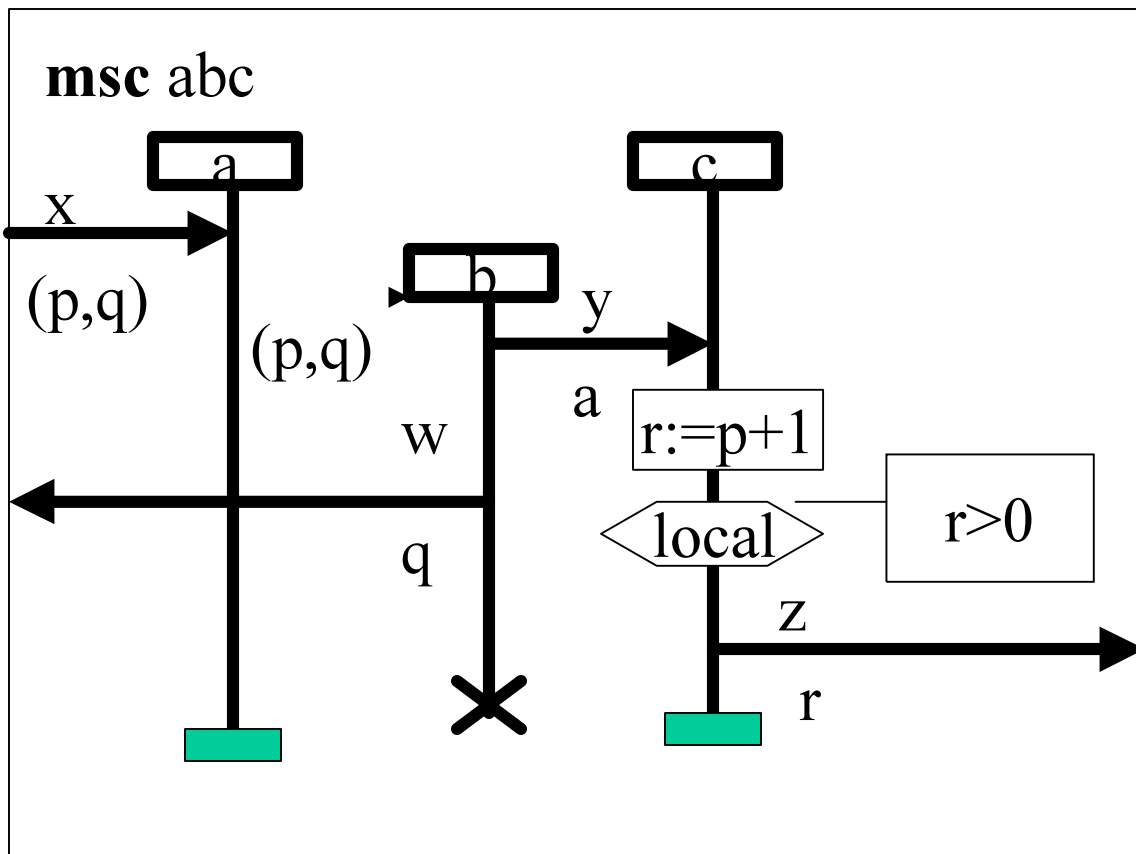
Local data flows

- actions
- information passing between actors
- local conditions

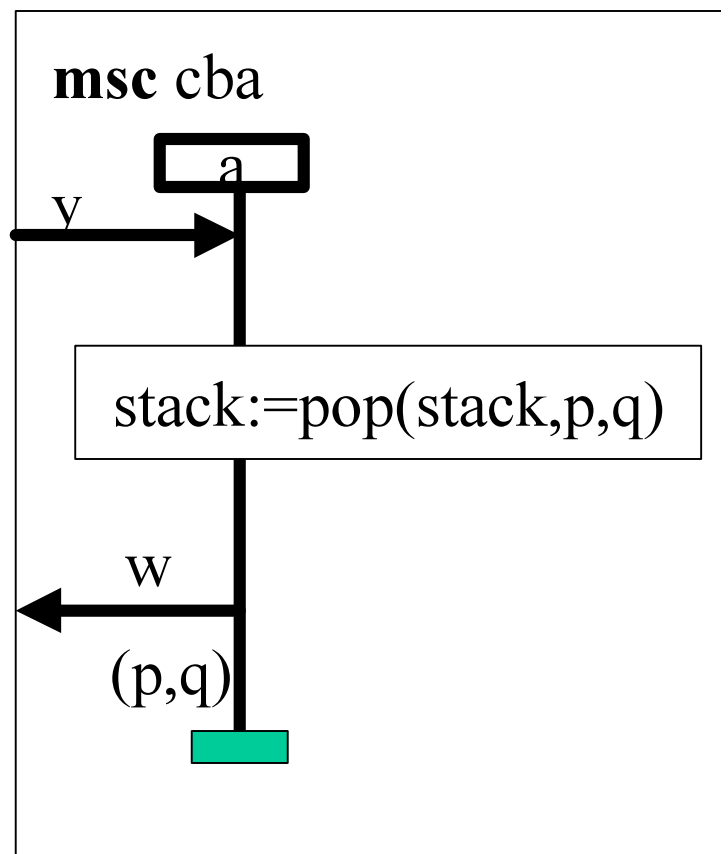
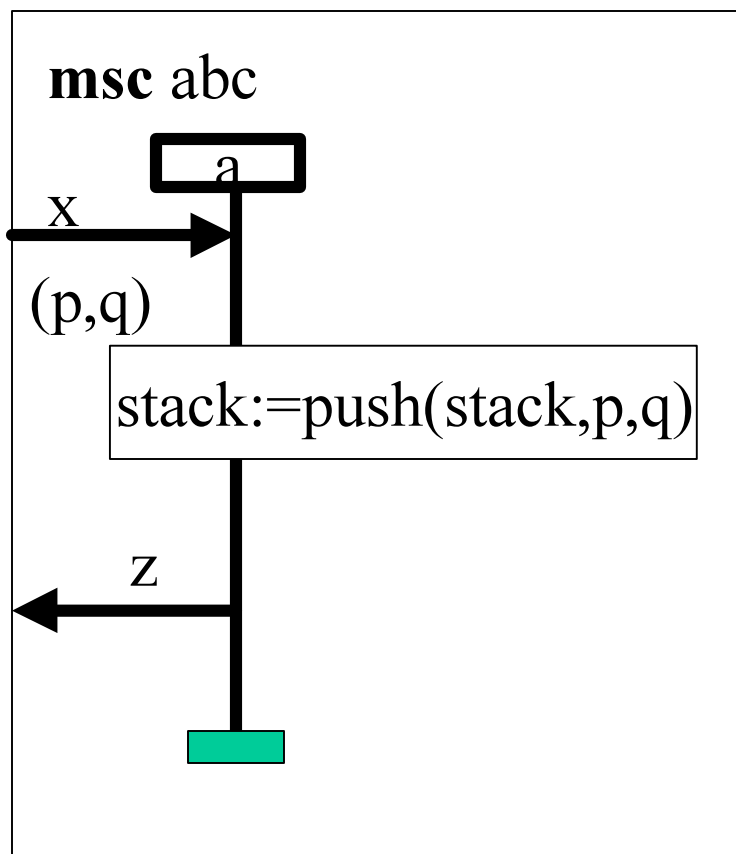
Data-flow relations between scenarios

- variables

Local data flow



Global data flow



Extensions to MSC

Actions

- `var := expr`
- `func(expr_1, ... ,expr_n)`

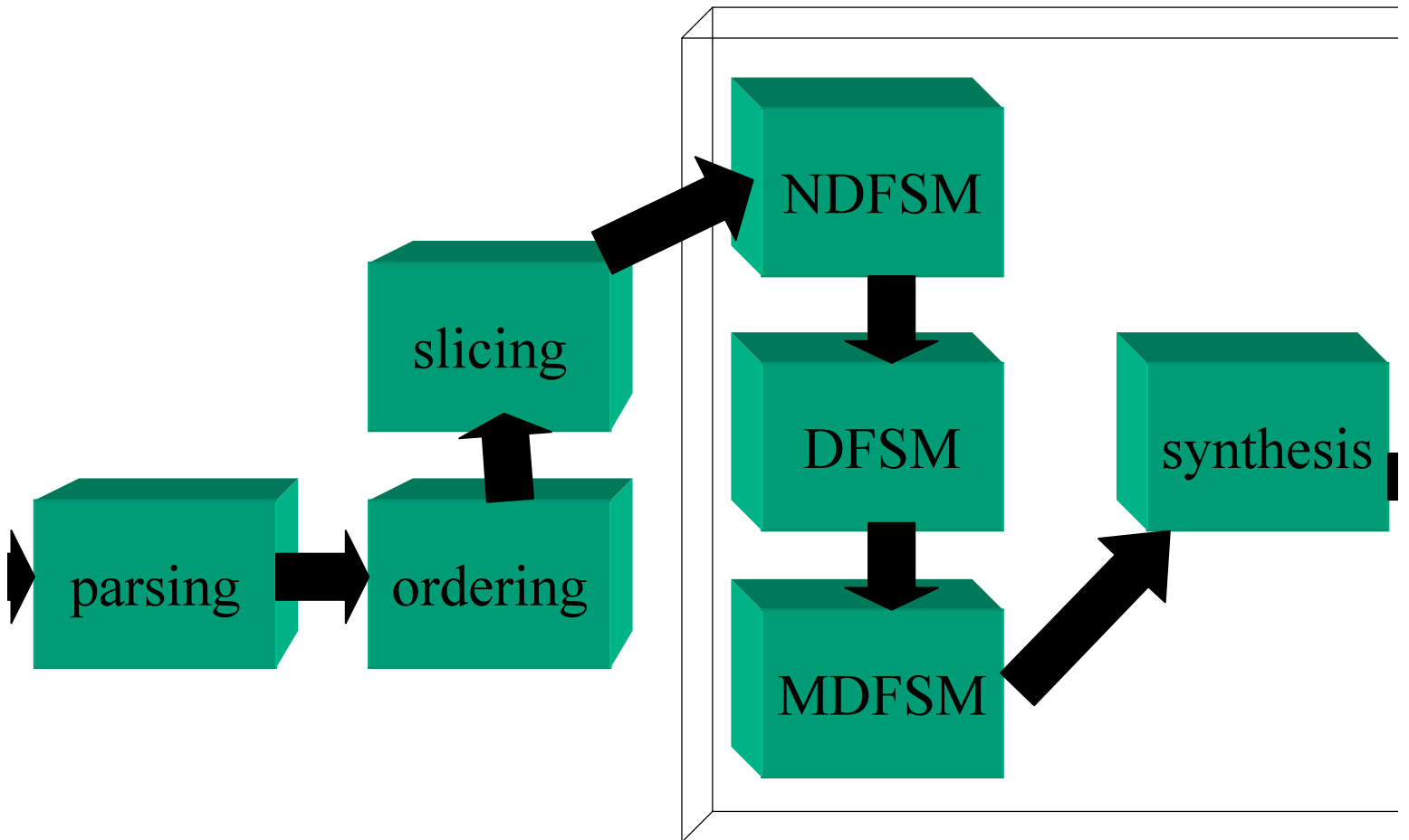
message parameters

- only variable names

local conditions with boolean expression as comment

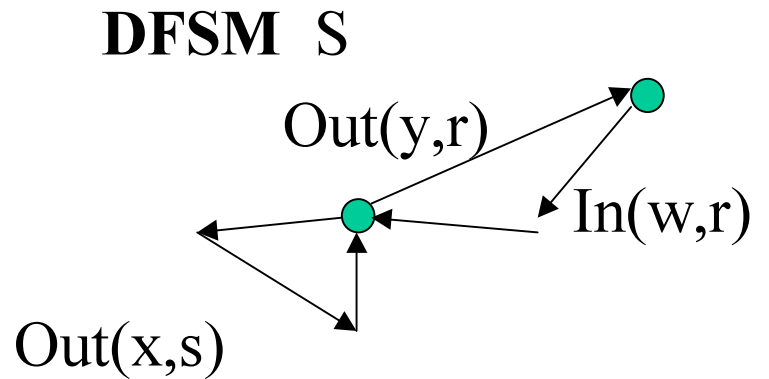
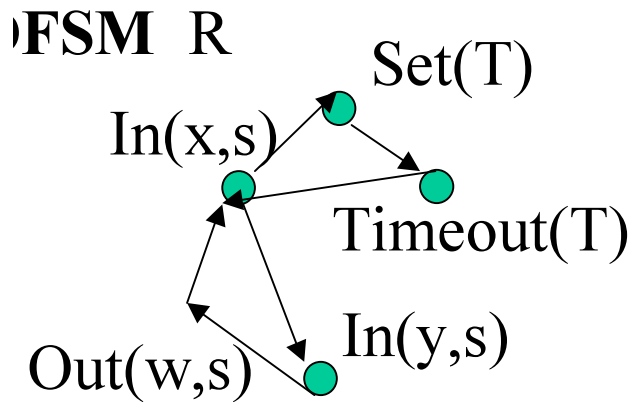
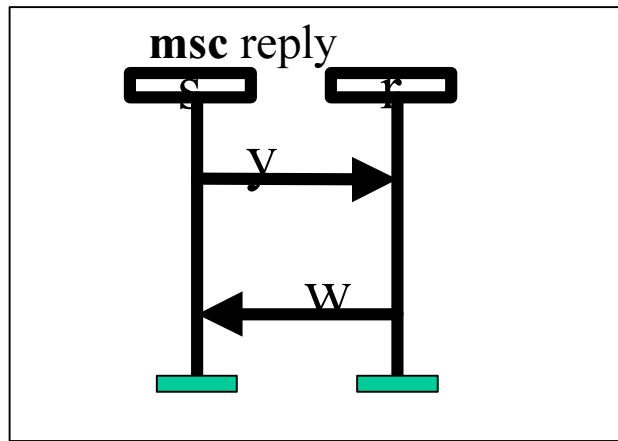
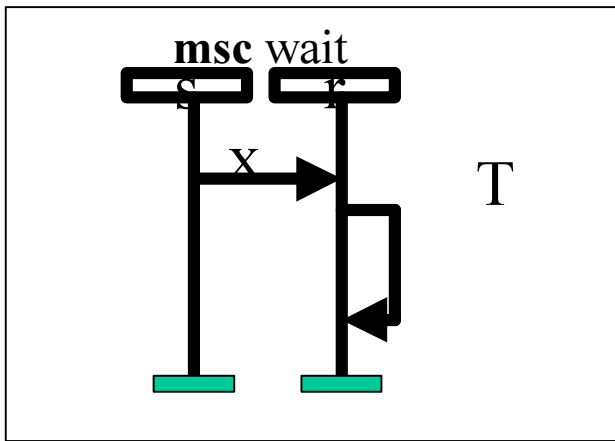
- `var <op> { var | const }`

Synthesis Algorithm



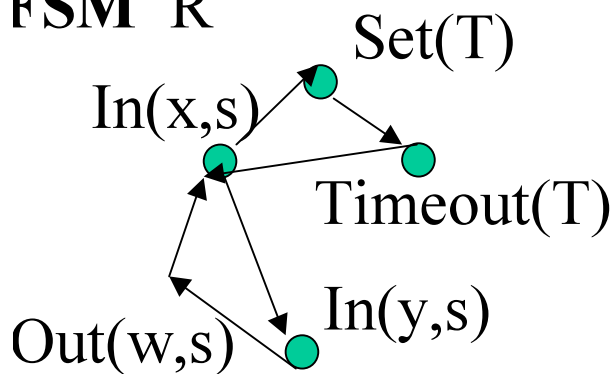
Dr. Nikolai N. Mansurov, Dept. of CASE tools, ISP

Synthesis Algorithm (2)

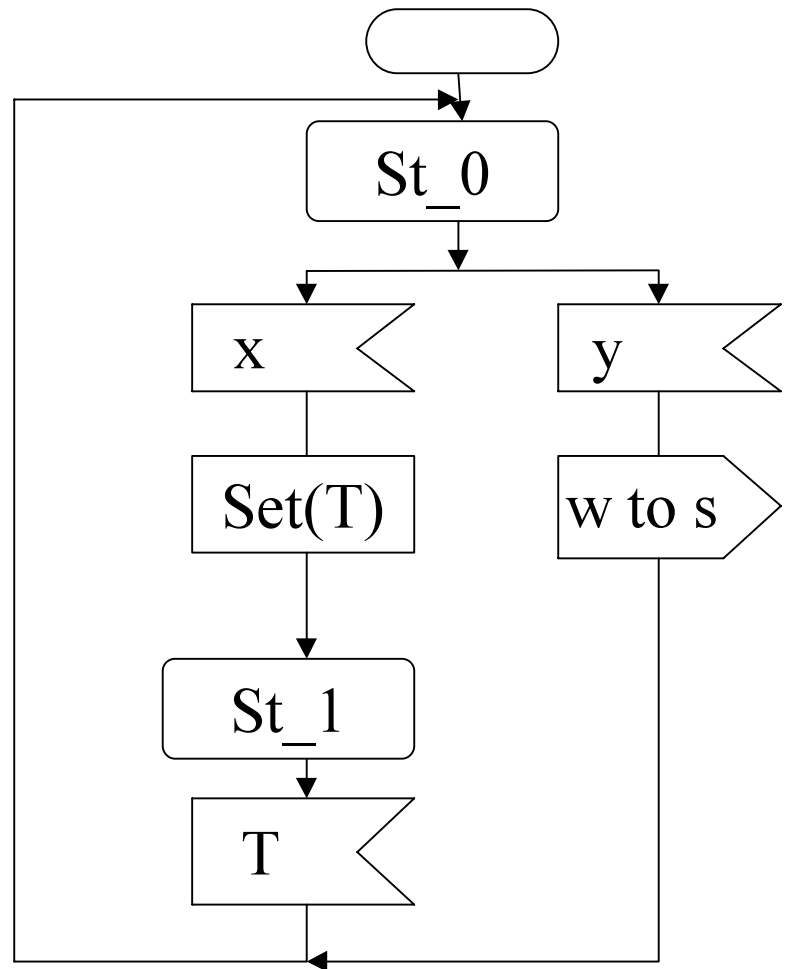


Synthesis Algorithm (3)

FSM R

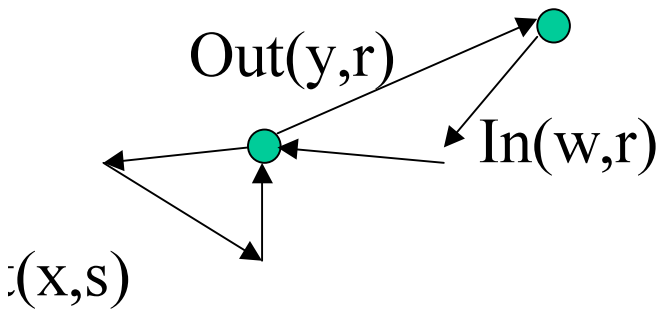


process R

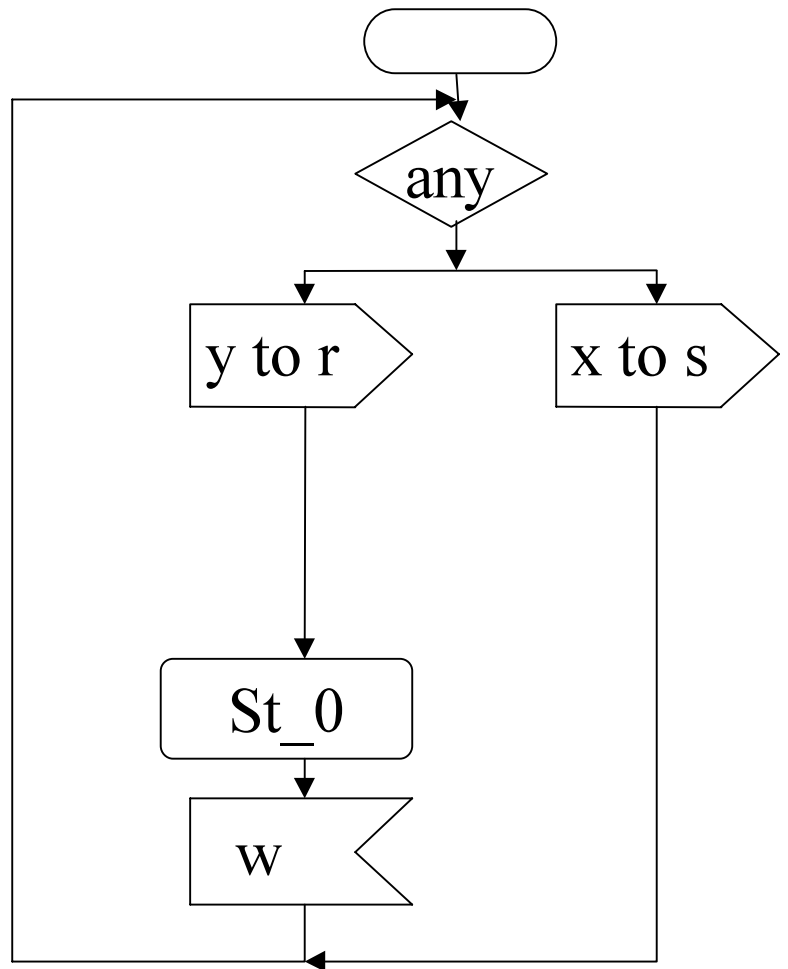


Synthesis Algorithm (4)

DFSM S



process S



Conclusions

Automatic synthesis of SDL models is suitable for rapid prototyping in MSC

Can be applied for re-engineering legacy software

Dramatically reduces learning curve for SDL