Лекция 8
Основни концепции на ориентирания към състояния изглед на системата

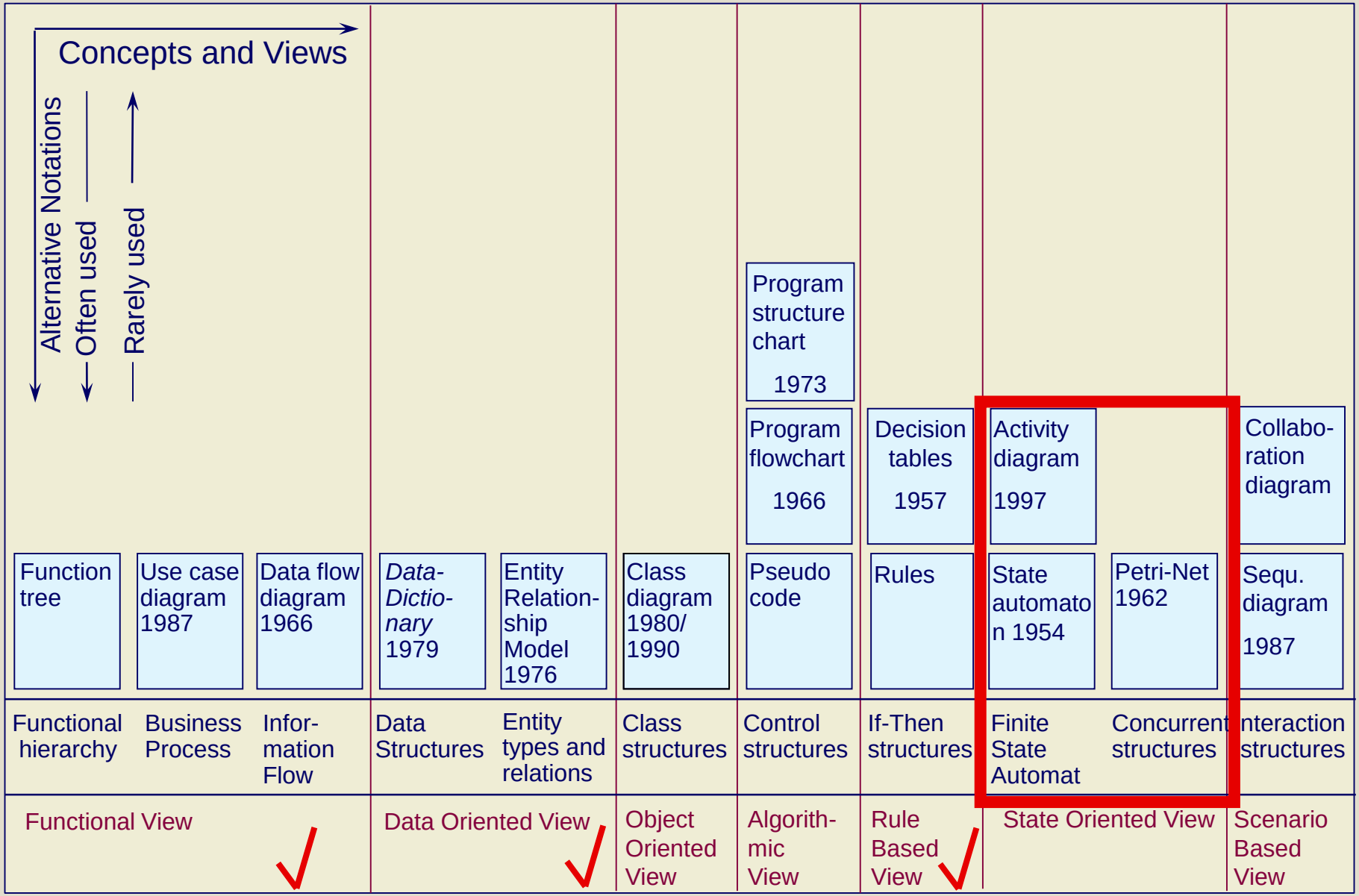
DAAD Project
“Joint Course on Software Engineering”

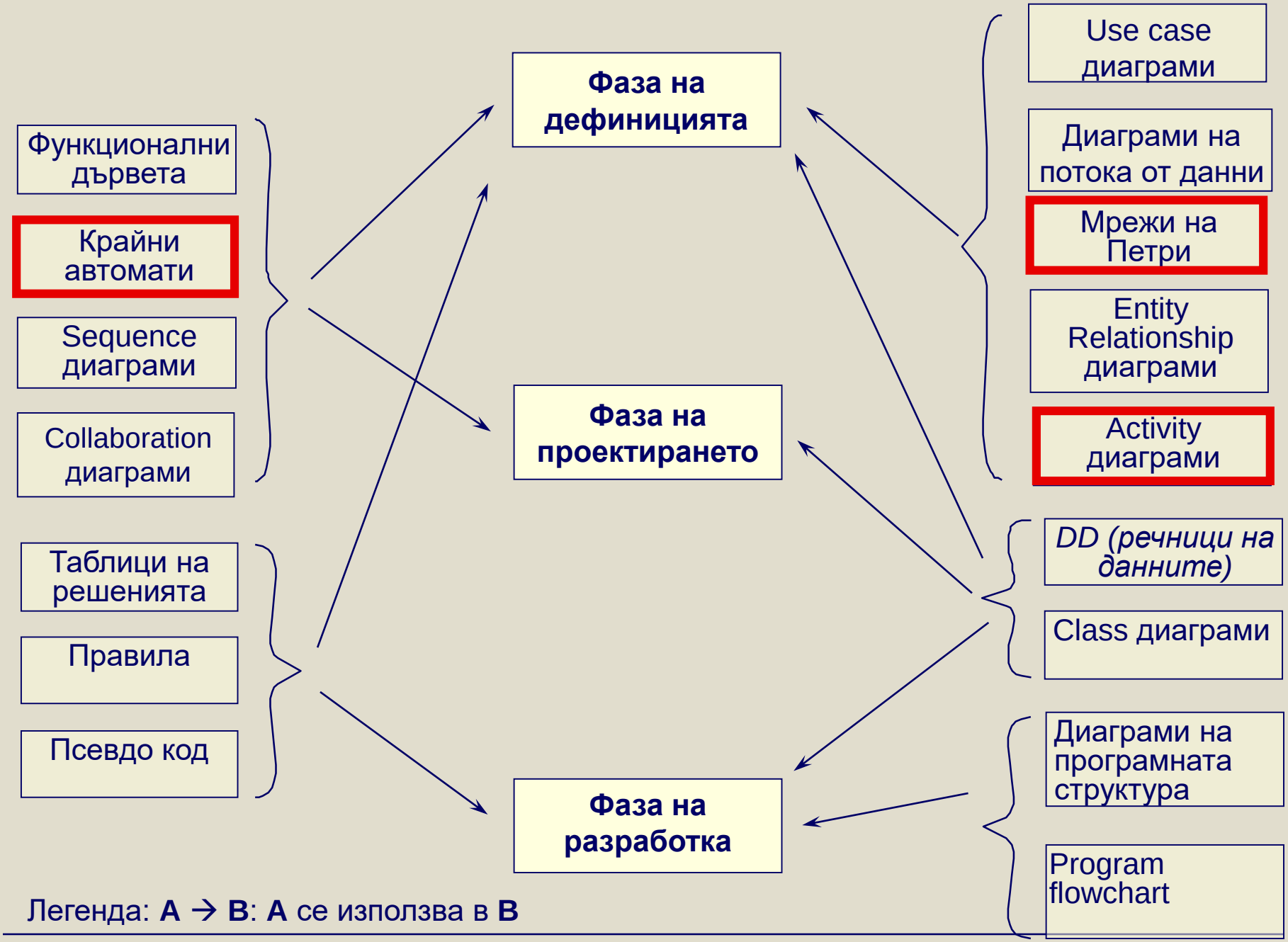
**Humboldt University Berlin, University of Novi Sad, University of Plovdiv,
University of Skopje, University of Belgrade, University of Niš, University of Kragujevac**

**Parts of this topic use material from the textbook
H. Balzert, “Software-Technik”, Vol. 1, 2nd ed., Spektrum Akademischer Verlag, 2001**

Version: Aug. 02, 2003 (D July. 25, 2003)

Основни концепции: ориентиран към състояния изглед на системата





Легенда: **A** → **B**: **A** се използва в **B**


графично
текстово
Граф на състоянията
Мрежи на Петри
ER
Дървета на решенията
Диагр. на потока от данни
Use Case диагр.
Class diagram
Диаграми на програмните структури
Activity диаграми
Функционални дървета
PAP
Sequence диагр.
езикова
Collaboration диаграми
Таблици на състоянията
Номерирани изисквания
ET
Крайни автомати (текстови)
Псевдо код
Мрежи на Петри(текст)
текст
Класификационни схеми
Правила
DD
неформални
полуформални
формализирани
формални

11. Основни концепции на ориентирания към състояния изглед на системата

a) Крайни автомати

b) Activity диаграми

Крайни автомати: цел и употреба

- **Цел:** Моделиране на поведението на системата, което зависи от вътрешното състояние на системата
  - Вътрешните състояния зависят от предишни въвеждания, т.е. история

- **Употреба:**
  - → динамичен изглед на обектите в класа: жизнен цикъл на обекта
  - → спецификацията на операциите на класа: промяна на състоянието, докато операцията се изпълни
  - → спецификация на use cases: промяна в състоянието по време на взаимодействието между потребител и система

Крайни автомати: дефиниция

- **Крайни автомати**:
  - Множество състояния (Z): крайно множество от вътрешни състояния
  - Множество входове (X): входове, сигнали, събития
  - Множество изходи (Y): изходи, действия
  - Преходи между състоянията: новото състояние зависи от входа и текущото състояние
  - Изходно поведение: изходите зависят от входа и текущото състояние

Модели

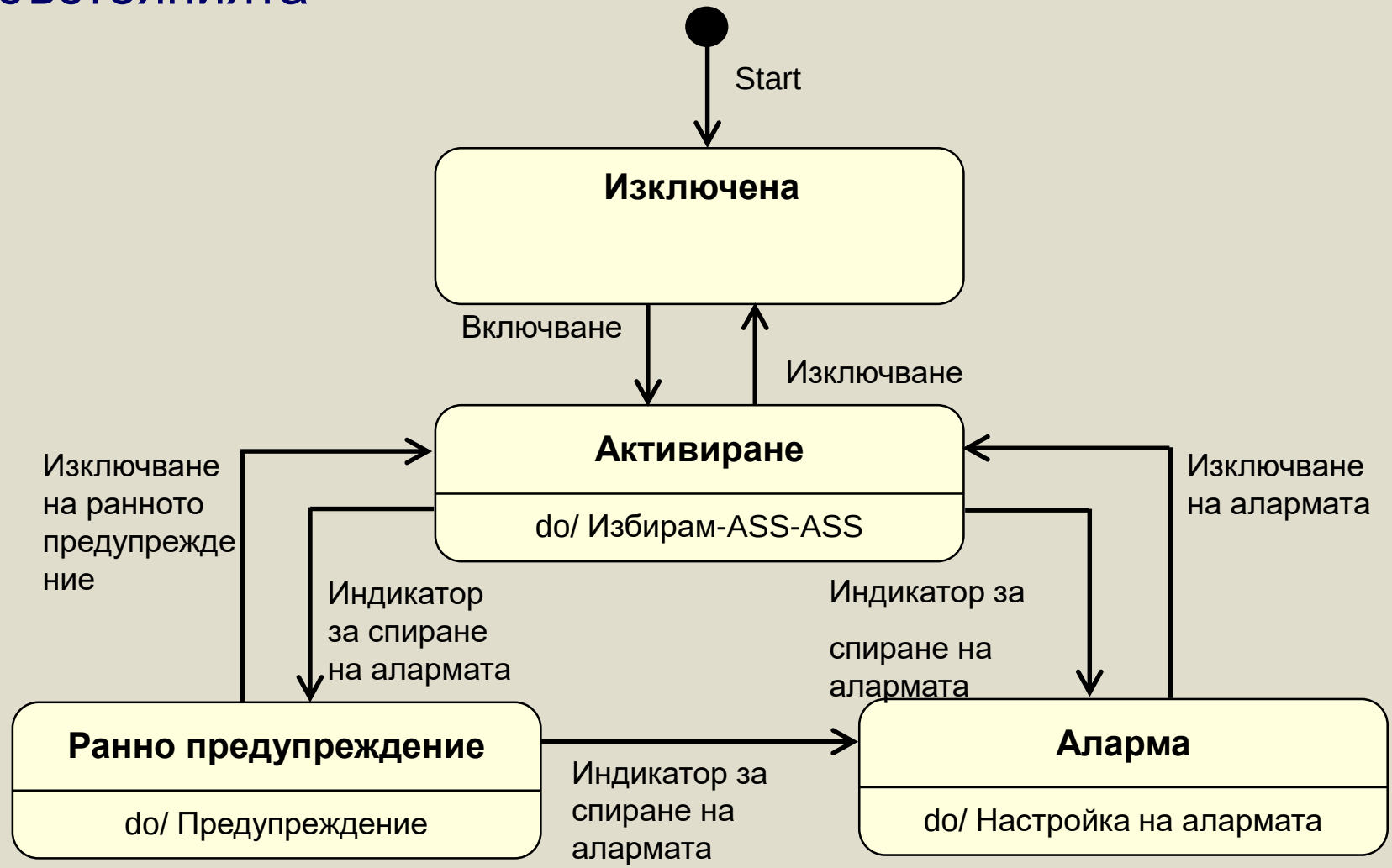
1. Математически модел: автомат = 5 елемента
   - A = [X, Y, Z, f, g] като
     - X, Y, Z са крайни състояния (вход/изход/състояния)
     - f: X x Z →Z (функция на преход)
     - g: X x Z →Y (функция на изхода)

2. Математически модел: Граф на състоянията

→ Автомат на Mealy, Автомат на Moore, Автомат на Harel (диаграми на състоянията)

Граф на състоянията: Настройка на аларма

- Нотация на Moore: Изход (действия) прикрепени към състоянията



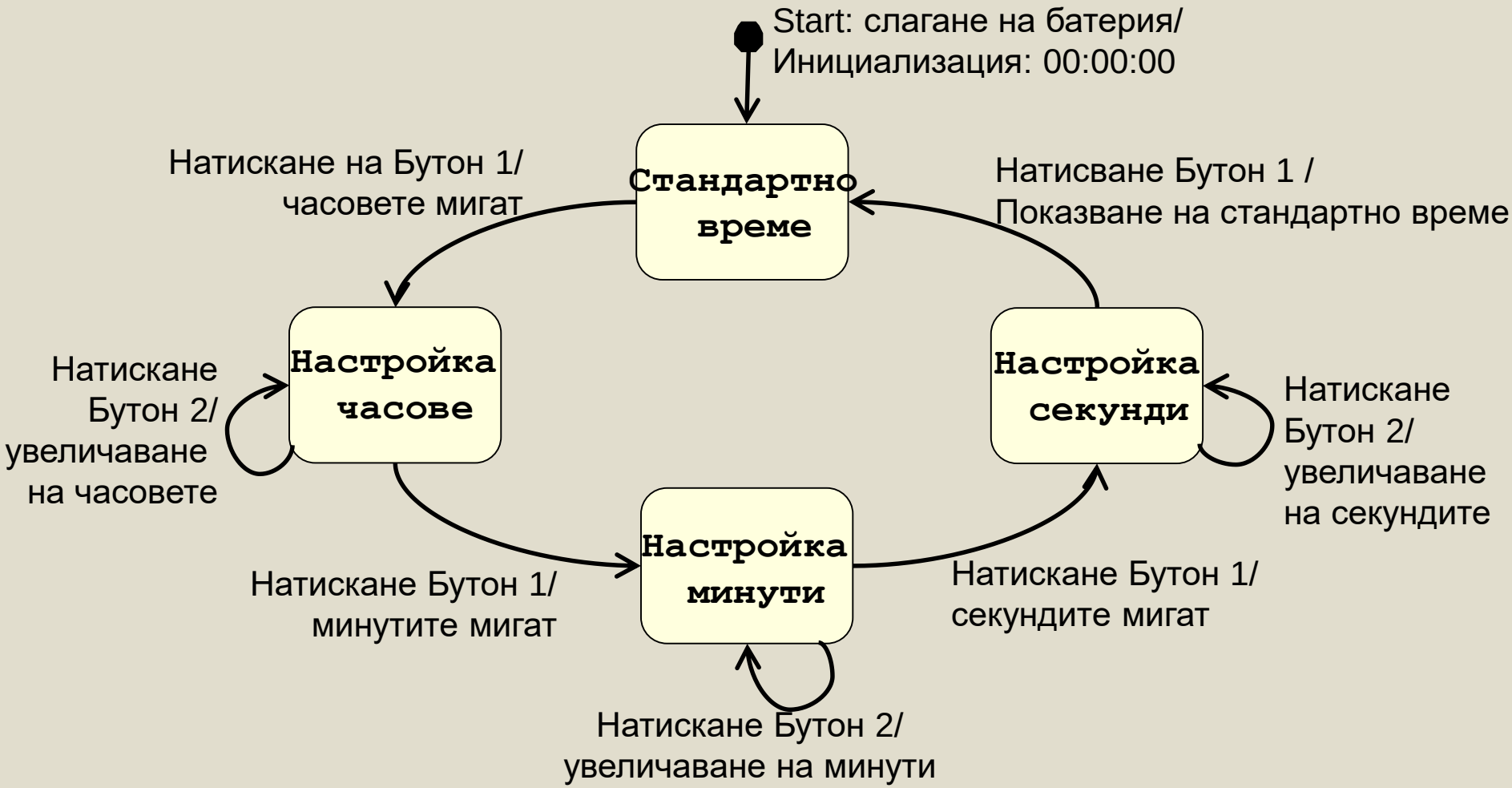
Пример: настройка на часовник

- Моделиране на настройването на дигитален часовник с два бутона за настройка
- *Бутон 1*
  - Позволява последователно избиране на настройка (стандартно време, настройка на часа, настройка на минутите, настройка на секундите)
- *Бутон 2*
  - Позволява настройка на времето според избраната настройка (стандартно време, час, минути, секунди)

Кой автомат ще се използва ?

Граф на състоянията: настройка на часовник

- Нотация на Mealy: Изходите/действията са прикрепени към преходите



Създаване на краен автомат(1)

- 1. Идентификация на *състоянията*, *входове/събития* и *изходи/резултати*
  - *Състояния*
    - **Състояние *Стандартно време*: След слагане на батерията**
    - **Състояние *Настройка на часове*: Часовете могат да бъдат настроени**
    - **Състояние *Настройка на минути*: Минутите могат да бъдат настроени**
    - **Състояние *Настройка на секунди*: Секундите могат да бъдат настроени**
  - ***Събития***
    - ***Начален сигнал***: Когато батерията е сложена
    - *Бутон 1 натиснат*: Когато бутон 1 е натиснат
    - *Бутон 2 натиснат*: Когато бутон 2 е натиснат
    - Два бутона не могат да бъдат натиснати едновременно

Създаване на краен автомат(2)

- *Резултати*
  - *Часовете мигат*
    - за да индицира, че в момента часовете могат да се настройват
  - *Минутите мигат*
  - *Секундите мигат*
  - *Увеличаване на часовете*
    - часовете се увеличават с по един на дисплея
  - *Увеличаване на минутите*
  - *Настройка на секундите*
    - 00 се показва на дисплея за секундите
  - *Инициализация*
    - Показва се на дисплея 00:00:00

Създаване на краен автомат(3)

- 2. Дефиниция на преходите
  - Когато *Началният сигнал* се чуе, системата трябва
    - да си смени състоянието на *Стандартно време* и
    - да изпълни действието *Инициализация*
  - Ако *бутон 1* е натиснат и състоянието на системата е *Стандартно време*, системата трябва
    - Да изпълни действието *Часовете мигат* и
    - Да промени състоянието си в *Настройка на часовете*
  - И така нататък…

Граф на състоянията: Нотация на Mealy

Настройка на часовник: няма крайно състояние?

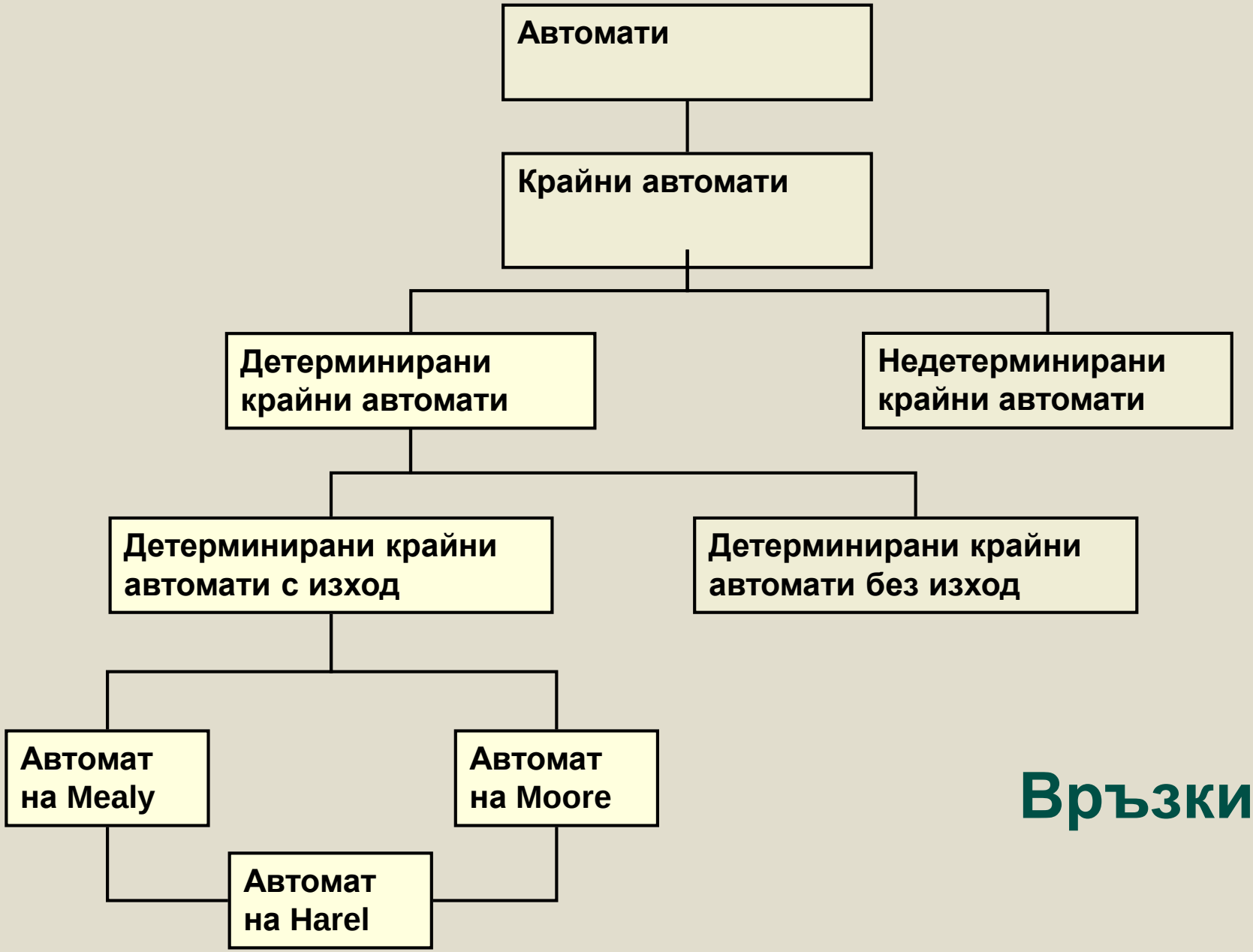
Алтернативна нотация: таблица на състоянията

- Необходима е когато графичното представяне не е достатъчно ясно
- Избягвайте използването на таблици ако е възможно

Текущо състояние	Събитие	Действие	Следващо състояние
	Старт	Инициализация	Стандартно време
Стандартно време	Бутон 1 натиснат	Часовете мигат	Настройка часове
Настройка на часове	Бутон 1 натиснат	Минутите мигат	Настройка минути
	Бутон 2 натиснат	Увеличаване на часове	Настройка часове
Настройка на минути	Бутон 1 натиснат	Секундите мигат	Настройка секунди
	Бутон 2 натиснат	Увеличаване на минутите	Настройка минути
Настройка на секунди	Бутон 1 натиснат	Показване на стандартно време	Стандартно време
	Бутон 2 натиснат	Настройка секунди	Настройка секунди

Автоматите на Mealy и Moore са еквивалентни

- Винаги могат да бъдат трансформирани един в друг
- Предпоставка за автомата на *Moore*
  - В съответно състояние има точно един изход и/или действие
- Автомата на Moore може да бъде трансформиран в еквивалентен автомат на Mealy
  - Изход генериран от състояние в автомата на Moore се поставя към прехода, който води към това състояние
- Автомата на Mealy може да бъде трансформиран в еквивалентен автомат на Moore
  - Целевите състояния с различни изходи се представят чрез няколко състояния в резултатния автомат на Moore


Автомати
Крайни автомати
Детерминирани крайни автомати
Недетерминирани крайни автомати
Детерминирани крайни автомати с изход
Детерминирани крайни автомати без изход
Автомат на Mealy
Автомат на Moore
Автомат на Harel

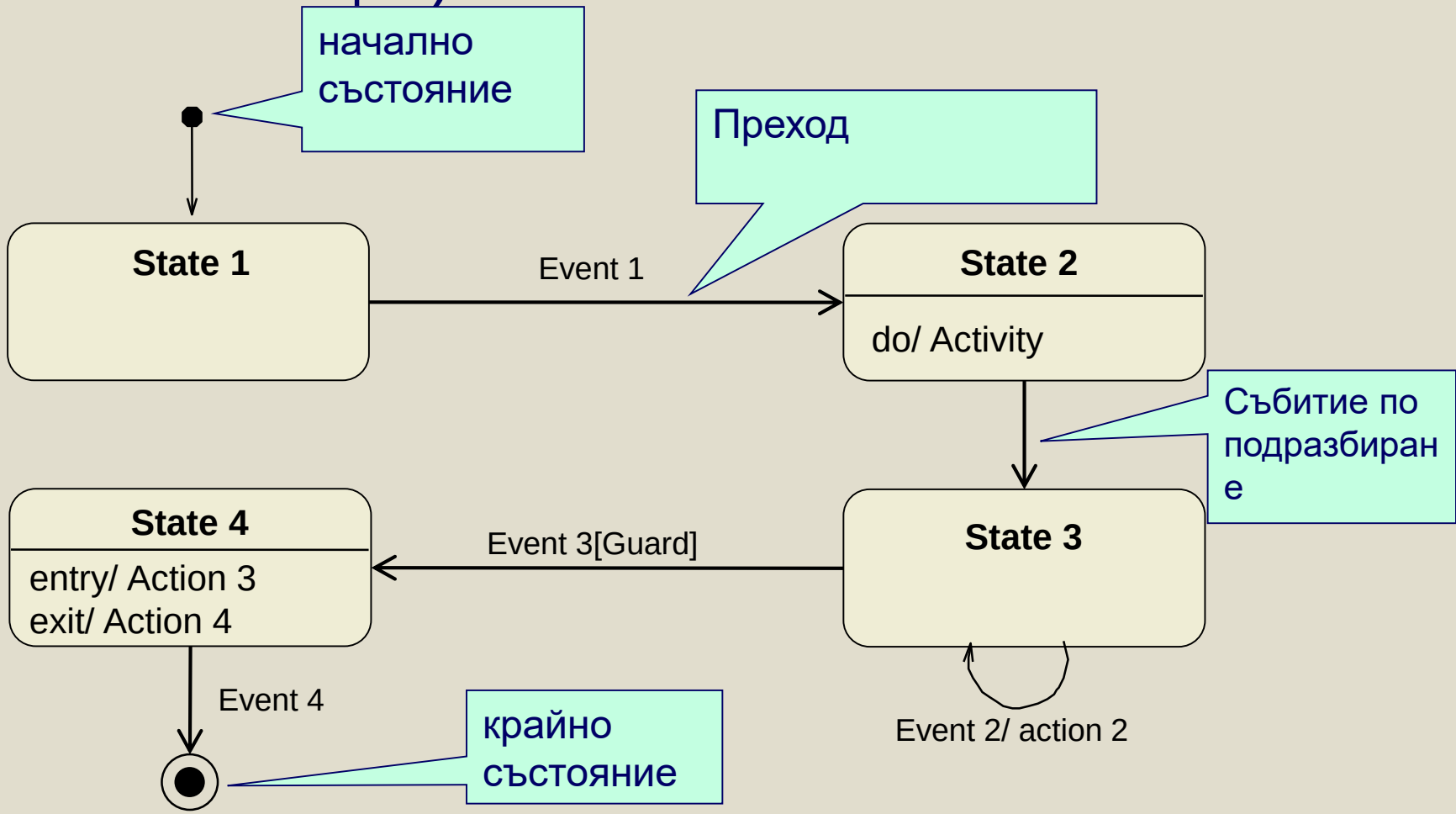
Връзки

Крайни автомати според Harel (statecharts)

- *Разширение* на въведената концепция:
  - *Нови понятия* за *автомата на Harel*
    - Хибридни крайни автомати ←
    - Условни преходи
    - Йерархични крайни автомати
    - Състояния с памет
    - Конкурентни състояния

Автомат на Harel: хибриден краен автомат

- Комбинация от автоматите на Mealy и Moore (в UML-нотация)



История

- Prof. Dr. David Harel
  *12.4.1950 в Лондон
  Професор по математика в института Weizmann в Rehovot, Израел
- Откривател на автоматите на Harel (*statecharts*) (1983, публикуван през 1987)
- Съразработчик на CASE системи за представяне на състояния (1984-1987, публикуван през 1988 и по-късно).

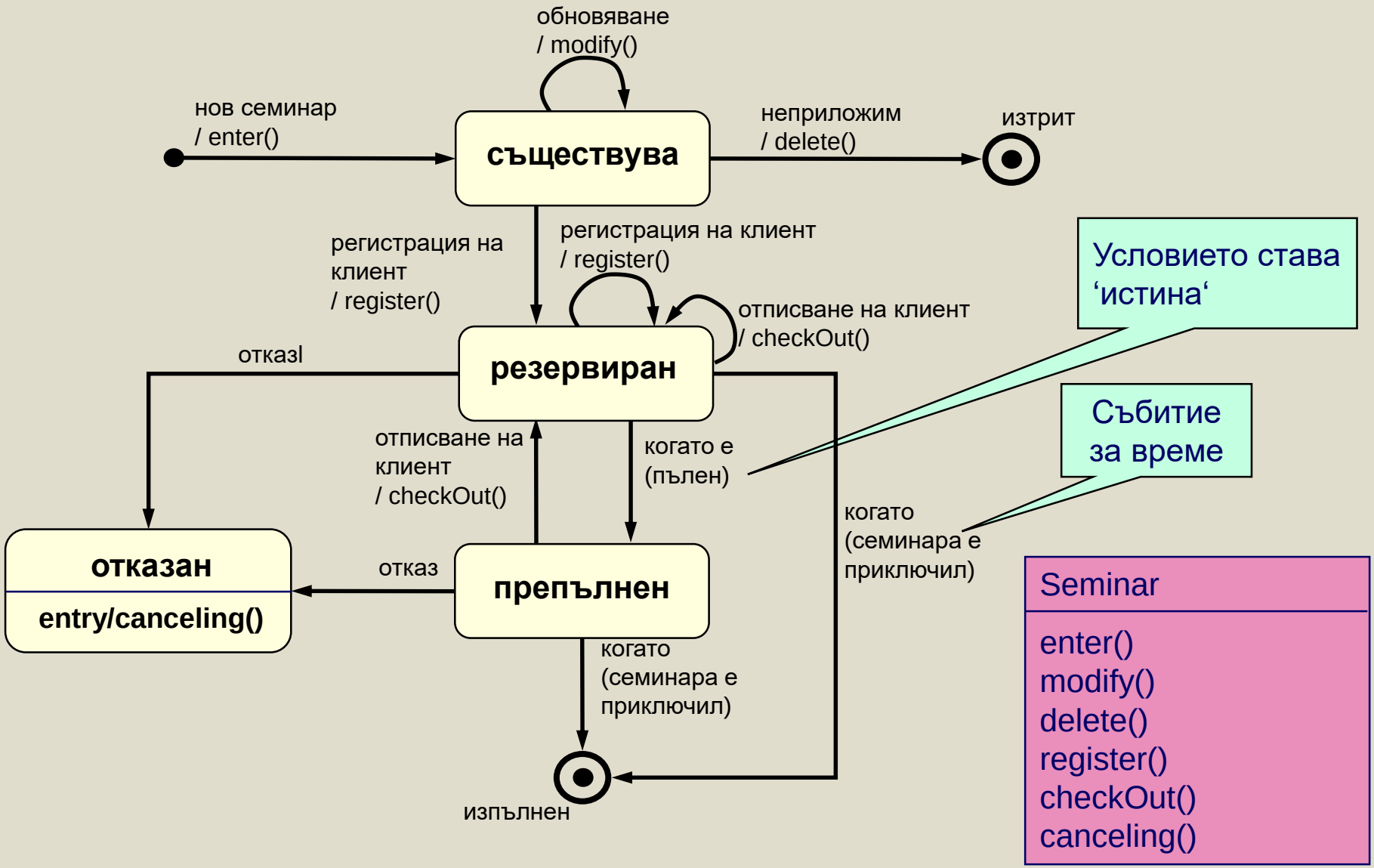
Крайните автомати в ОО свят

- **Крайни автомати...**
  - ... най-често се използват за моделиране на жизнения цикъл на обектите
  - Всички обекти в класа могат да се представят с крайни автомати
  - Всеки обект може да има индивидуално състояние
  - По принцип не е необходимо да се създава краен автомат за всеки клас.

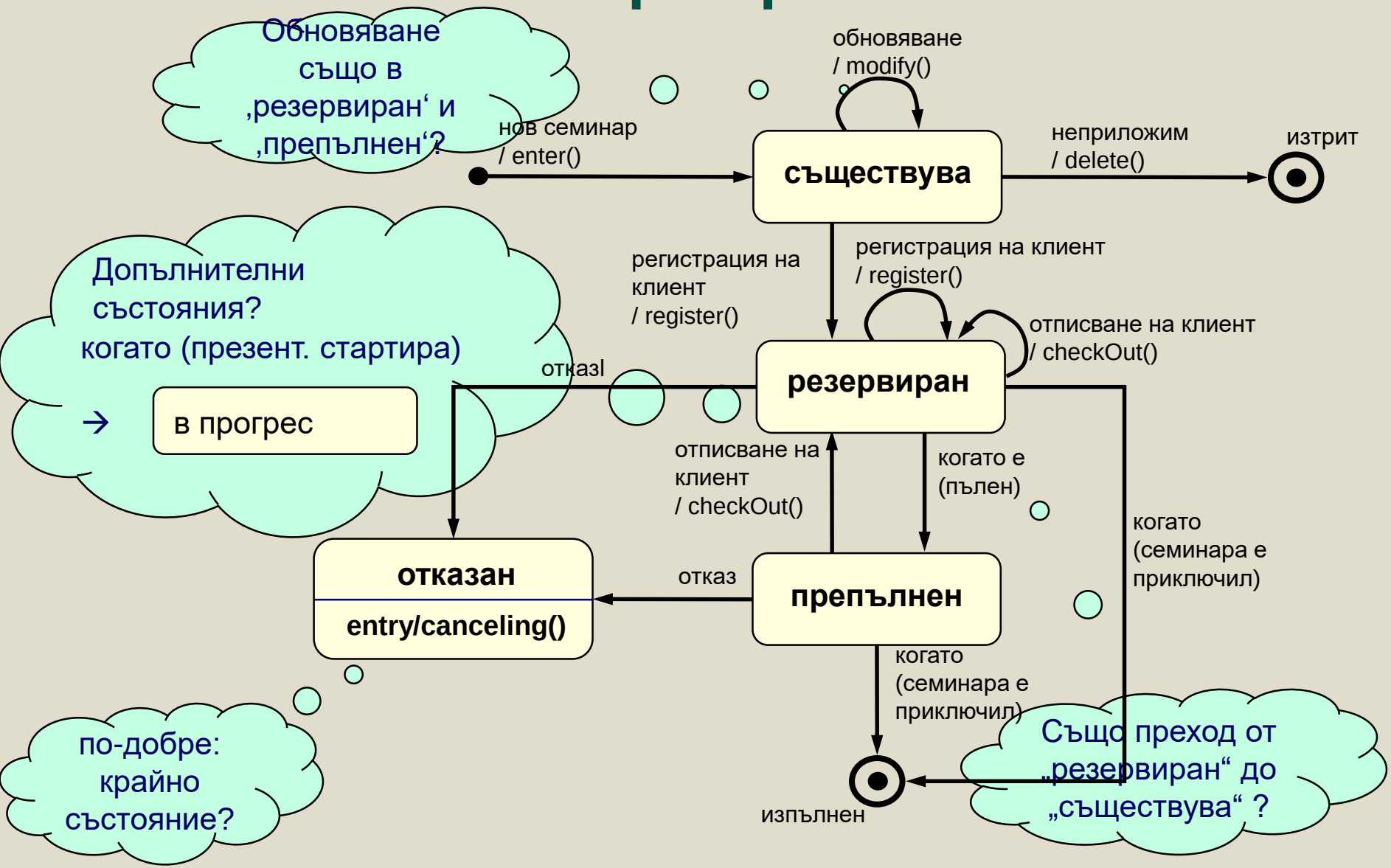
  → *динамичен модел* в ООА

- **Жизнения цикъл на обекта и клас диаграмите**
  - Само операциите на съответен клас се позволява да бъдат дейности

Жизнен цикъл на обект от клас ‘Семинар‘



Жизнен цикъл на обект от класа ‘Семинар‘: проблеми



Събития (events)

- *Събитие (event)* може да бъде:
  - условие, което става истина
  - сигнал
  - съобщение (извикване на операция)
  - изминало време или
  - събитие в определено време

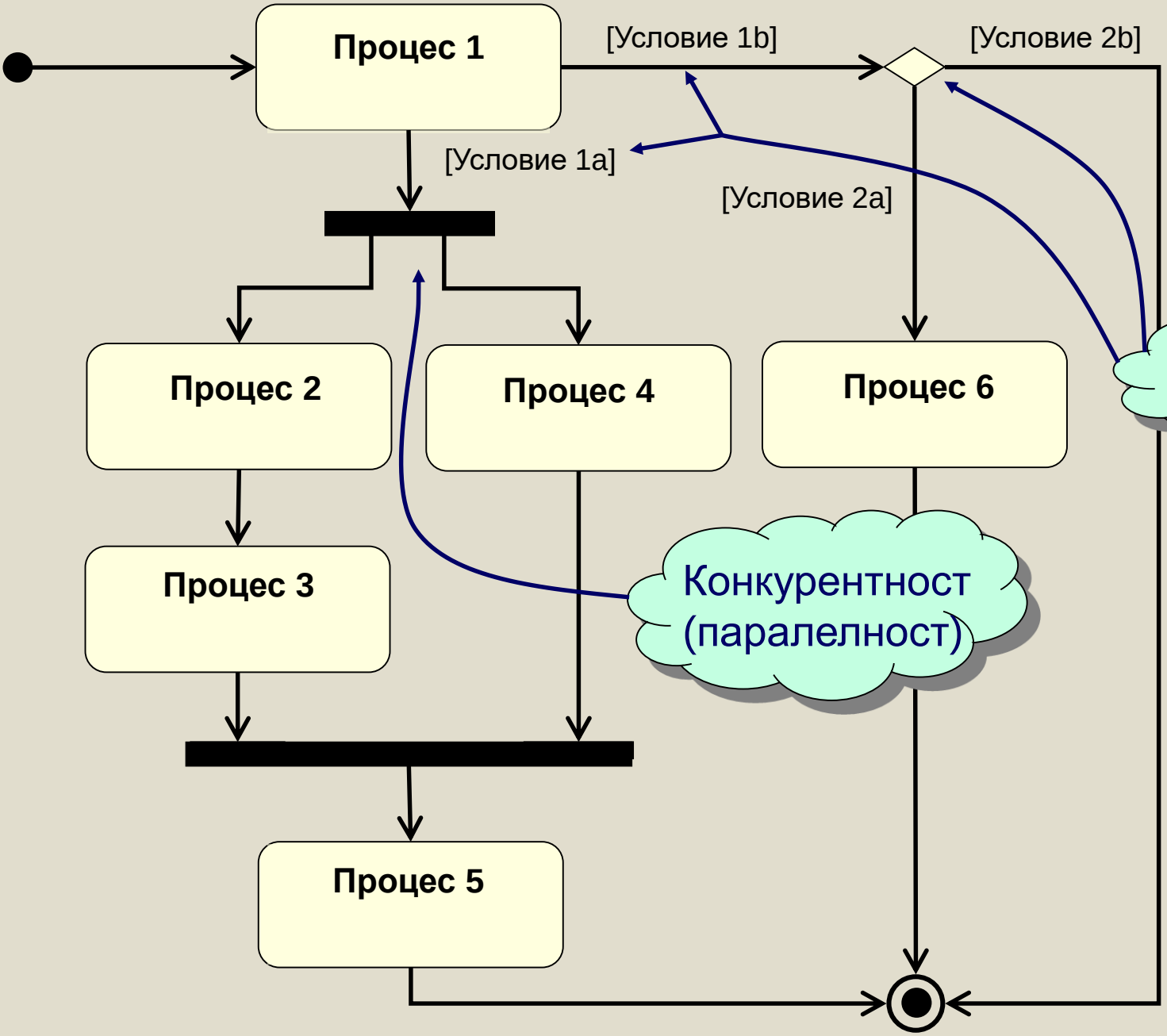
  - Последните два случая се отнасят за временни събития

11. Основни концепции на ориентирания към състояния изглед на системата

a) Крайни автомати

b) Activity диаграми

Activity диаграми

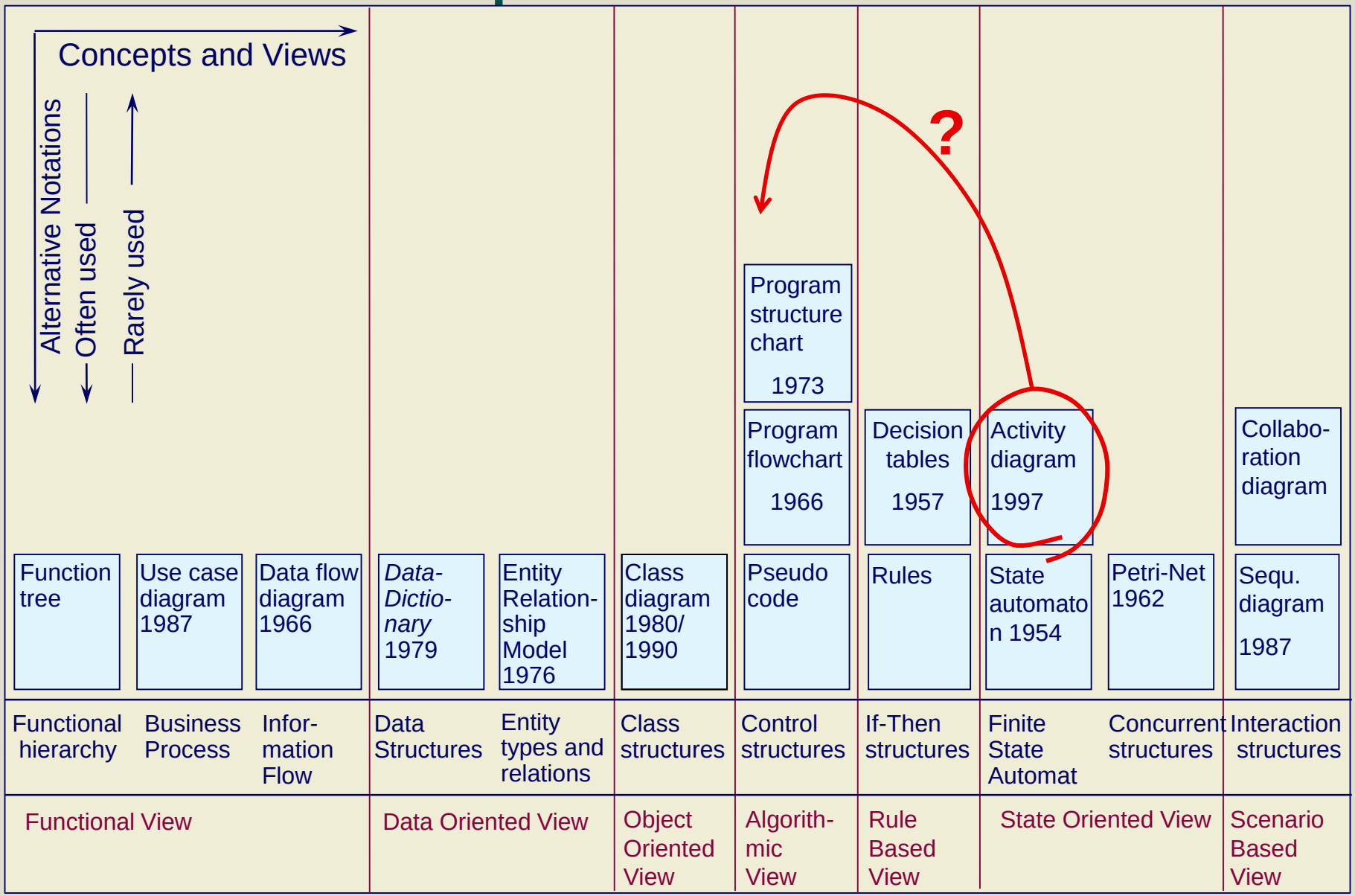


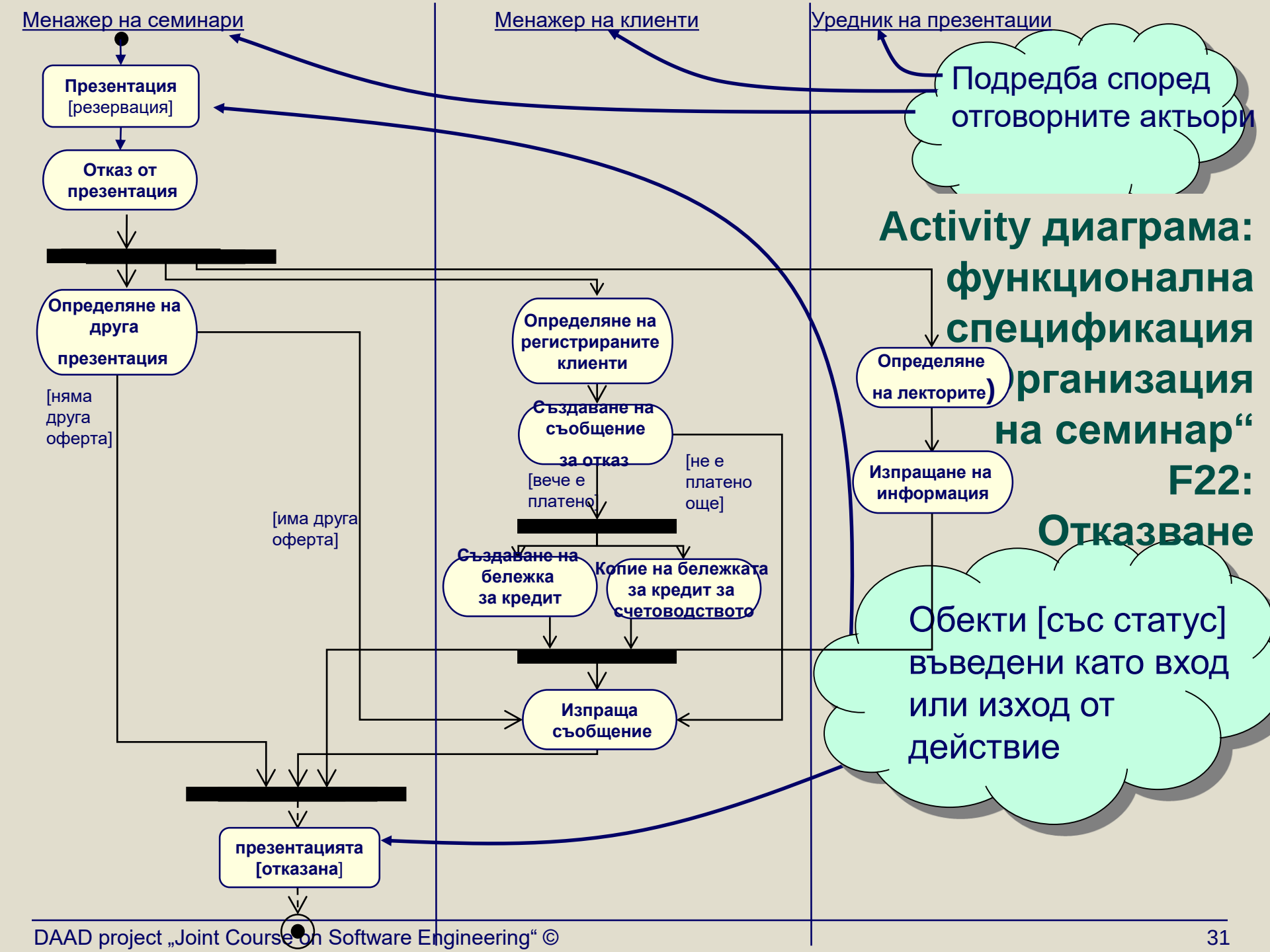
Последователност от дейности: Разклонения и конкурентност

Activity диаграми

- Те представят вариант на крайните автомати
- *Activity диаграми* – Описание на алгоритми и /или бизнес процеси чрез състояния на действията (action states)
- *Action state* = Стъпка (дейност) по време на изпълнението на алгоритъм или бизнес процес
- *Action state* е напуснато, когато дейността свързана с него е приключила (implicit event)
- Подобни на "старите" диаграми на потока и/или диаграми на програмния поток

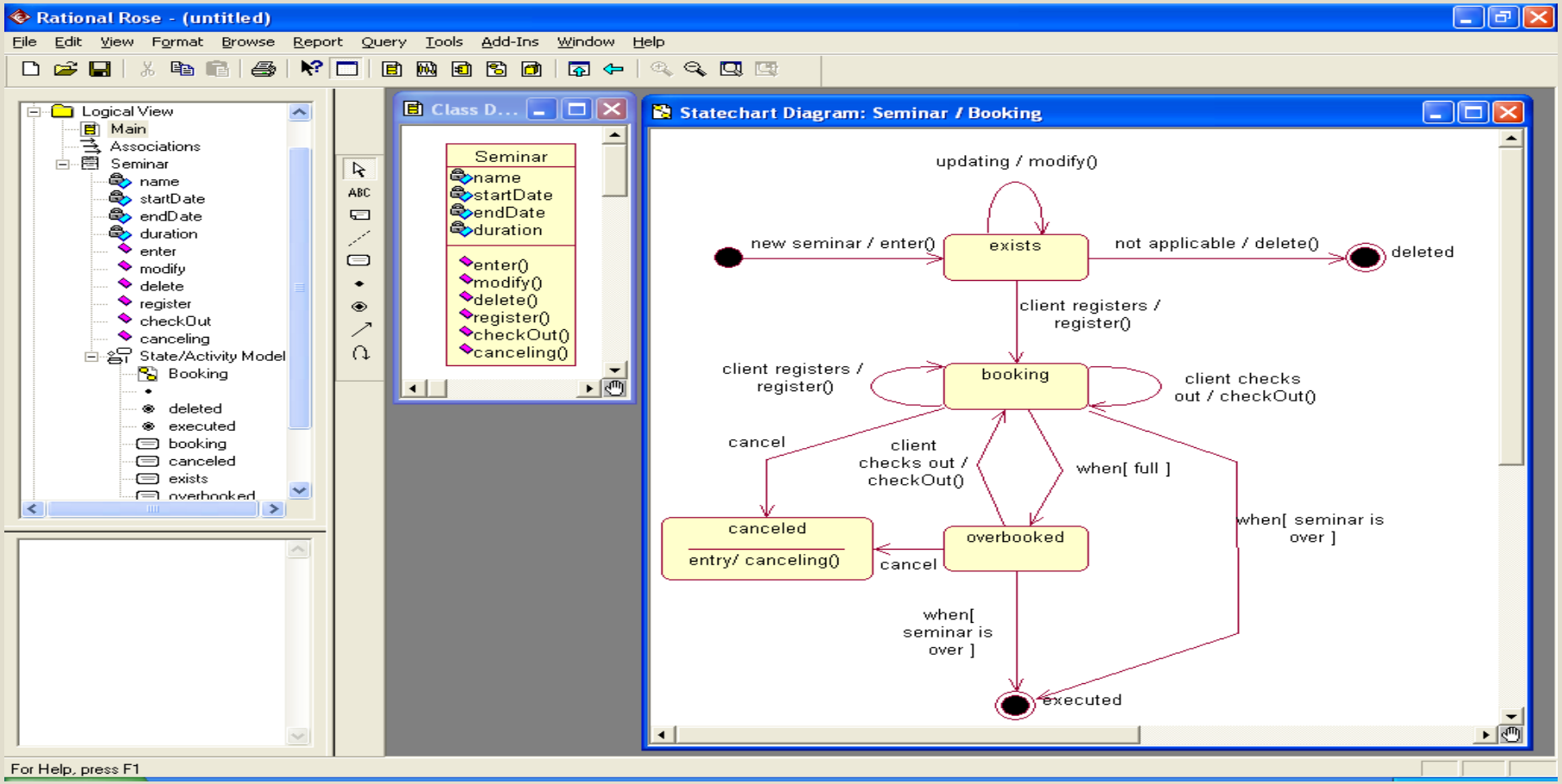
Класификация на Activity диаграмите. Правилна ли е?

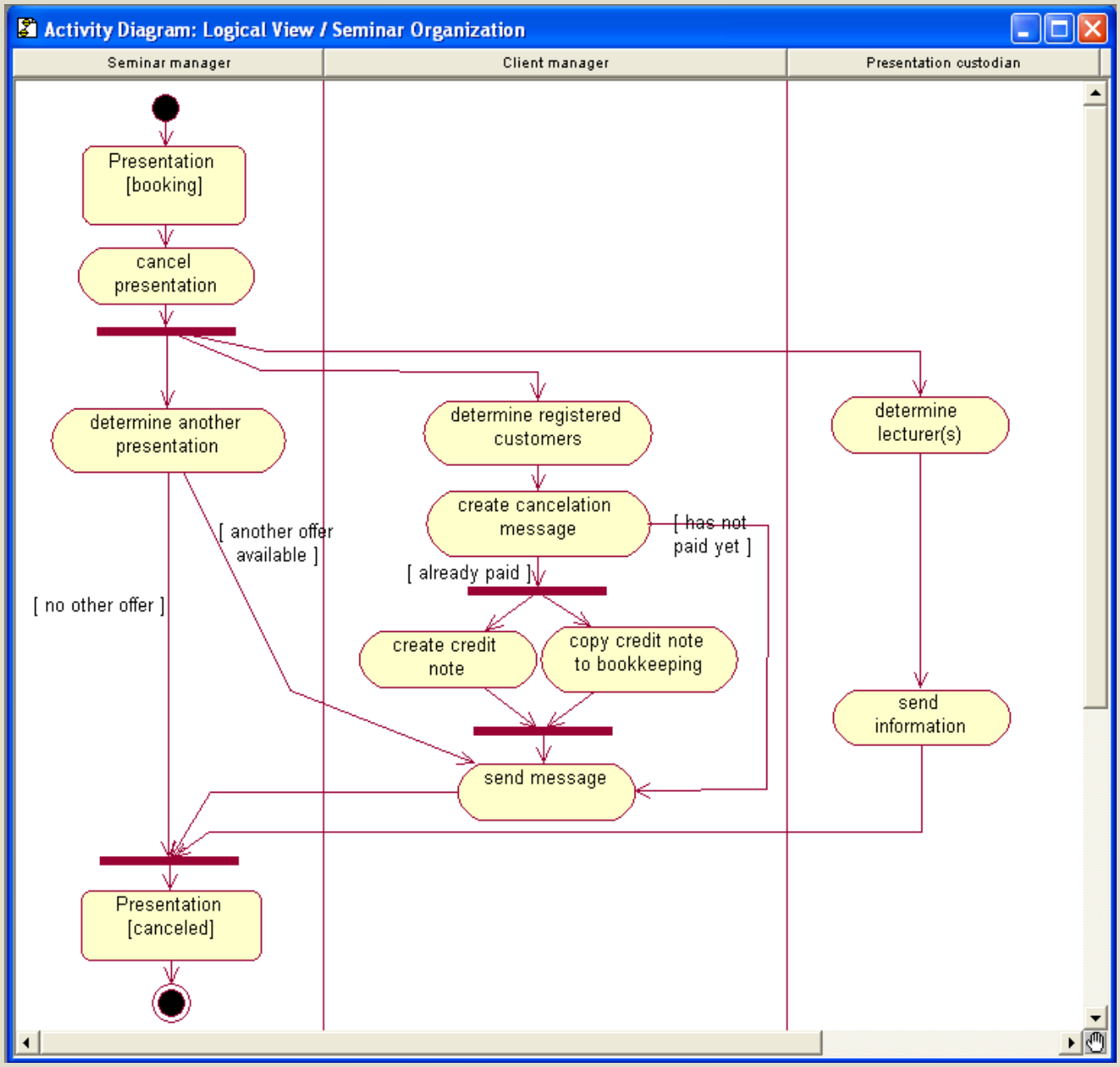



Менажер на семинари
Менажер на клиенти
Уредник на презентации
Подредба според отговорните актьори
Презентация [резервация]
Отказ от презентация
Определяне на друга презентация
[няма друга оферта]
[има друга оферта]
Определяне на регистрираните клиенти
Създаване на съобщение за отказ
[вече е платено]
[не е платено още]
Създаване на бележка за кредит
Копие на бележката за кредит за счетоводството
Изпраща съобщение
Определяне на лектории
Изпращане на информация
презентацията [отказана]
Activity диаграма: функционална спецификация „Организация на семинар“ F22: Отказване
Обекти [със статус] въведени като вход или изход от действие

Използване на CASE средства

- Класа Семинар и свързания с него краен автомат
  - Rational Rose





CASE: Activity диаграма на операцията ‘Отказване’