

# АНАЛИЗ НА СОФТУЕРНИТЕ ИЗИСКВАНИЯ

---

**Методи за извличане на изискванията,  
основаващи се на методи на различните  
гледни точки**

**(Viewpoint-oriented requirements methods)**

Лекция 9

# Съдържание

---

- Понятие за гледни точки при ИИ
- Понятие за гледни точки в структурния анализ
- Подходи на гледните точки за ИИ
  - SADT
  - CORE
  - VOSE
  - VORD
- Валидиране на изискванията чрез методи на гледните точки

# Анализ на изискванията, основаващ се на подход на гледни точки

---

- Извличането на изискванията включва идентифициране, анализ и (раз)решаване на *различни идеи, перспективи и взаимовръзки* на отделни нива на детайлност.
- Методите за анализ, които са базирани на строги общи схеми, *не могат да отчетат всички съществуващи различни знания*, които са съществени за анализа.
- *Решение:* Методи, базирани на идеята за *гледните точки*

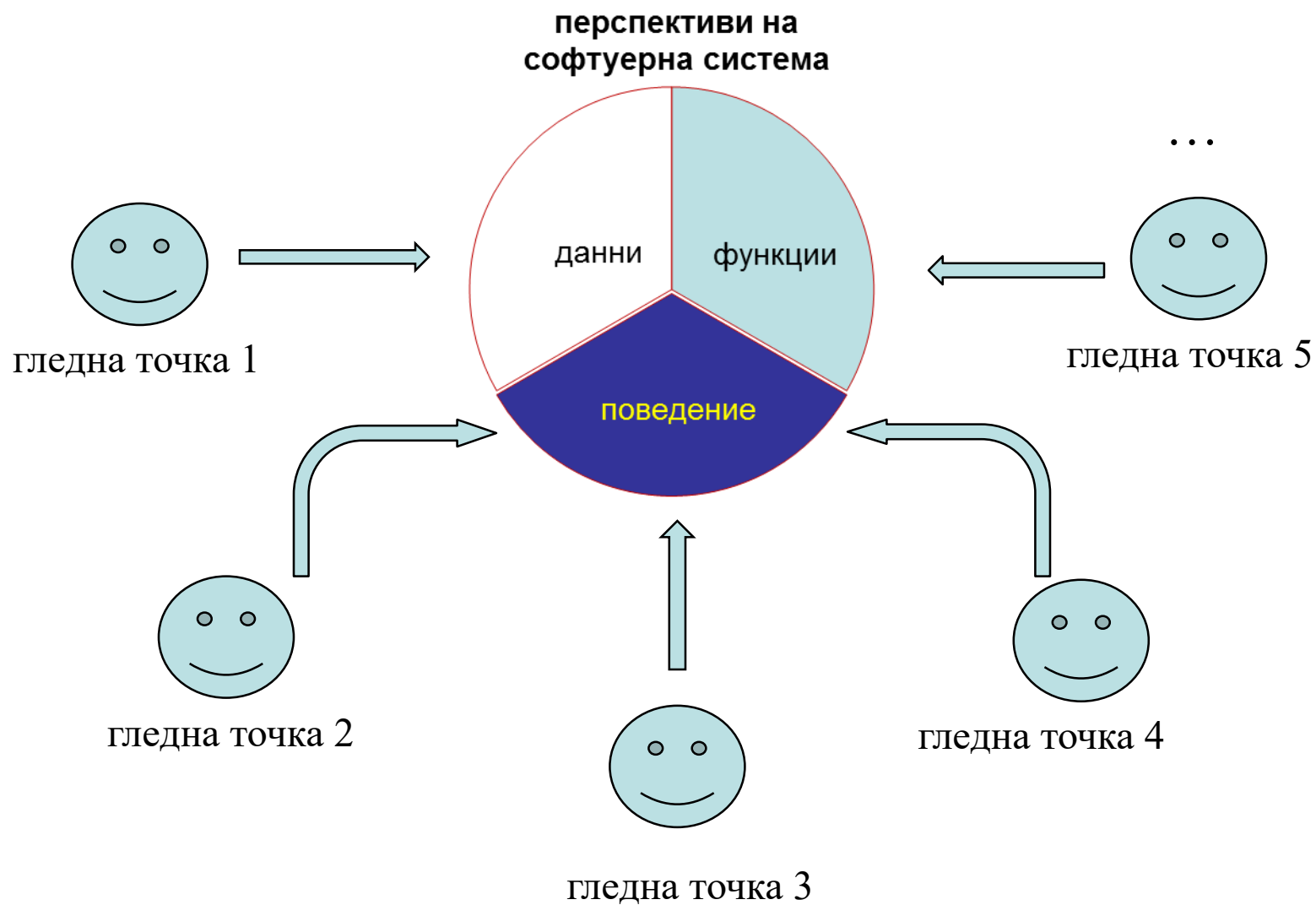
# Гледна точка (в контекста на ИИ)

---

*Дефиниция:* Гледна точка е съвкупност от информация за системата или свързан проблем, среда или област, която е събрана от **специфичното разбиране** (виждане) на краен потребител или на друга система, или на разработчик, или друго заинтересовано лице.

## *Въпроси:*

- Интегриране на информацията на отделните гледни точки;
- Решение на съществуващи конфликти



# ***Пример:***

---

Автоматична система за спиране, инсталирана във влак.

- Някои **гледни точки** и съответните изисквания са:
  - *Изисквания на машиниста (от какъв вид са?)*
  - *Trackside equipment (оборудване по трасето)*
  - *Инженерът по безопасността (диспечер)*
  - *Съществуващи системи във влака (какво се изисква?)*
  - *Характеристики на спирачките на влака*

## Предимства на подходите, базирани на гледни точки

---

- Разпознават *явно* разнообразието на източниците на изискванията.
- Осигуряват *механизъм* за организация и структуриране на тази *разнообразна* информация.
- Осигуряват *цялостност*.
- Осигуряват средство на източниците на изискванията или на заинтересованите страни *да открият и да проверят* своя принос към формулиране на изискванията.
- *Ефективност* на процеса на ИИ.

# Подход на гледни точки чрез SADT (1)

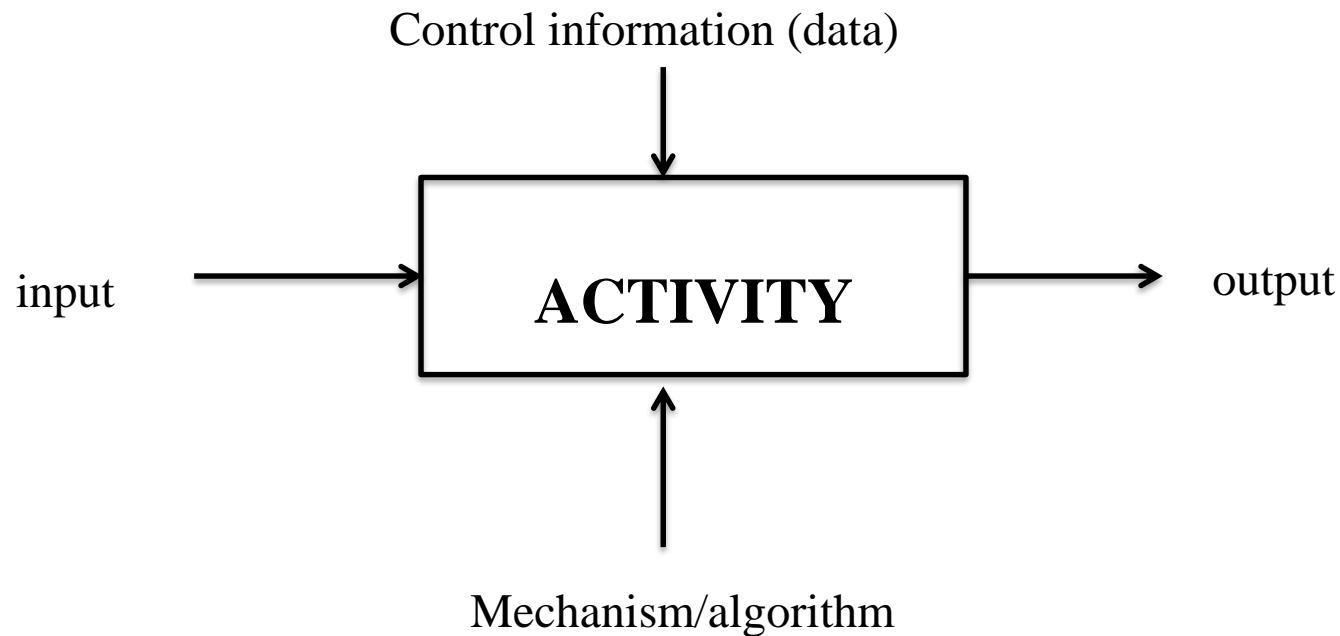
---

- Техниката за *структурен анализ* и *техника за проектиране* (**SADT**) е разработена в края на 80-те от Ross и е имала широко приложение.
- Използва нотацията на структурния анализ (напр. модела на потока на данните (DFD)) за описание на множество от взаимодействащи си дейности.



# SADT нотация

---



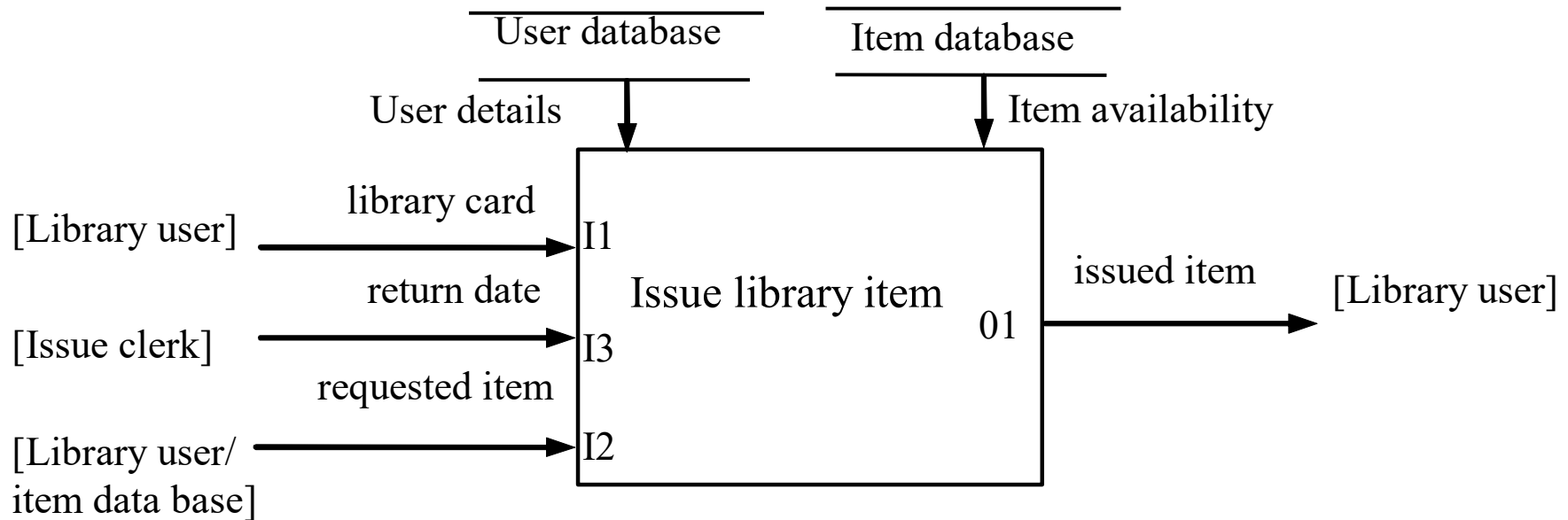
**The next level of decomposition needs as a starting point for functional requirements.**

## Подход на гледни точки чрез SADT (2)

---

- “*Гледните точки*” на SADT са различните *източници* и *приемници* (*sinks*) на данни.
- SADT **декомпозира** проблема в множество от **йерархично** свързани диаграми;
- SADT **не дефинира** явно гледните точки  
Гледните точки са **интуитивно разширение** на използваната от SADT техника за моделиране.

# Пример за библиотека



“Гледните точки” са показани в квадратни скоби

- Следващото ниво на декомпозиция свързва входната информация с управлението и изходната информация на по-детайлното ниво.
- Декомпозицията се повтаря до пълно изясняване на детайлите.

# Controlled Requirements Expression (CORE)

---

- CORE е разработен за British Aerospace в края на 70-те от system designers (Mullery, 1979). Използва се и в Европейската космическа програма след 1980 г., също European Fighter Aircraft.
- Методът CORE е базиран на подход за *функционалната декомпозиция*.
- CORE е базиран на *явно* дефиниране на гледни точки за формулиране на изискванията.

## Стъпки на CORE метода

---

- Методът CORE се състои от 7 итеративни стъпки:
  - Определяне на гледните точки
  - Структуриране на гледните точки
  - Таблично обединяване (Tabular collection)
  - Структуриране на данните
  - Отделно моделиране на гледните точки
  - Комбинирано моделиране на гледните точки
  - Анализ на ограниченията

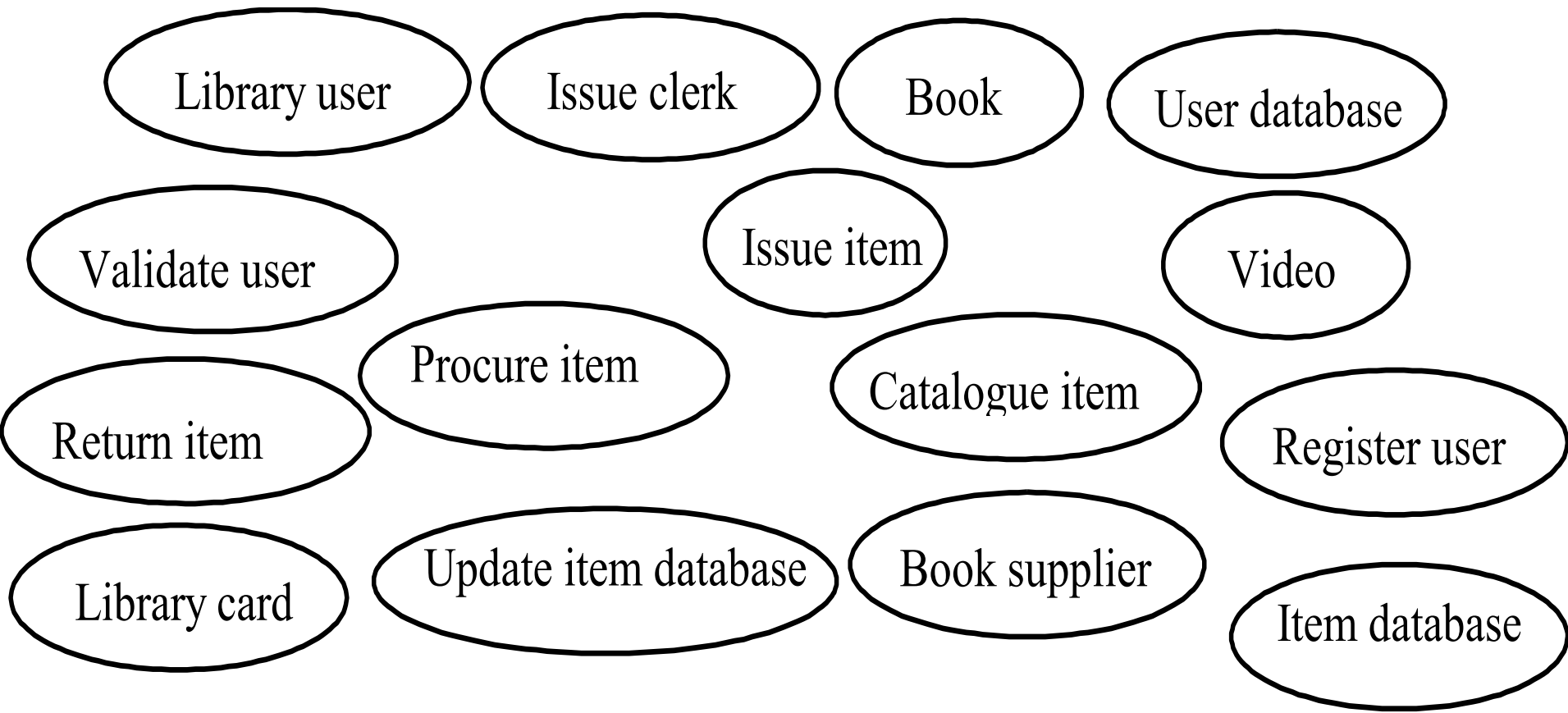
# Стъпка 1 – Определяне на гледните точки (1)

---

- *Определяне* на възможните гледни точки на **две** стъпки:
  1. Обхваща *всички обекти*, които взаимодействат със системата за идентифициране на функционални и нефункционални изисквания.
    - Няма строги правила за идентифициране на гледните точки;
    - „Brainstorming“ сравнение на мненията м-у отделни аналитици
  2. Прави разграничение между *defining (определящи)* и *bounding (гранични)* гледни точки

# Пример за библиотека – „първоначални“ гледни точки

---



## Стъпка 1 – Съкращаване на гледните точки (2)

---

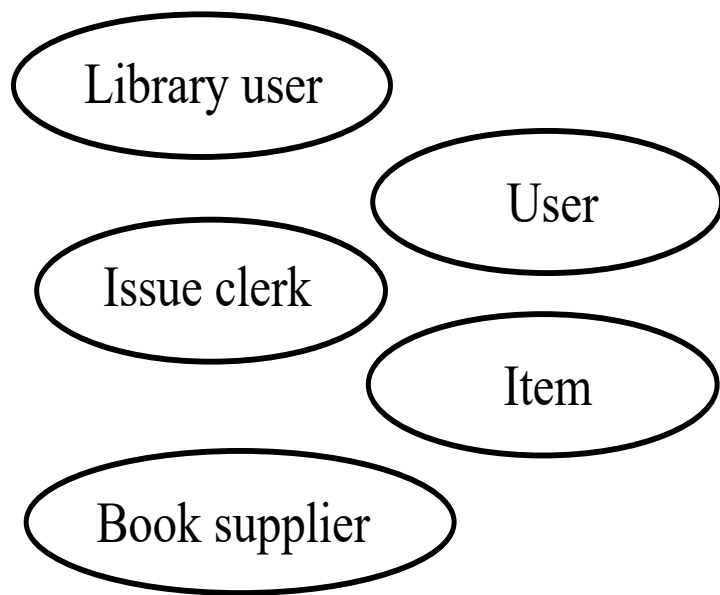
- Всяка елипса представя *най-абстрактната* форма на гледните точки
- Последният етап на идентификация на гледните точки включва съкращаване на идентифицираните гледни точки до *набори* от:
  - *Bounding (гранични) гледни точки*: Обекти, които взаимодействат със системата.
  - *Defining (определящи) гледни точки*: (Под)процеси на системата, разглеждани низходящо.



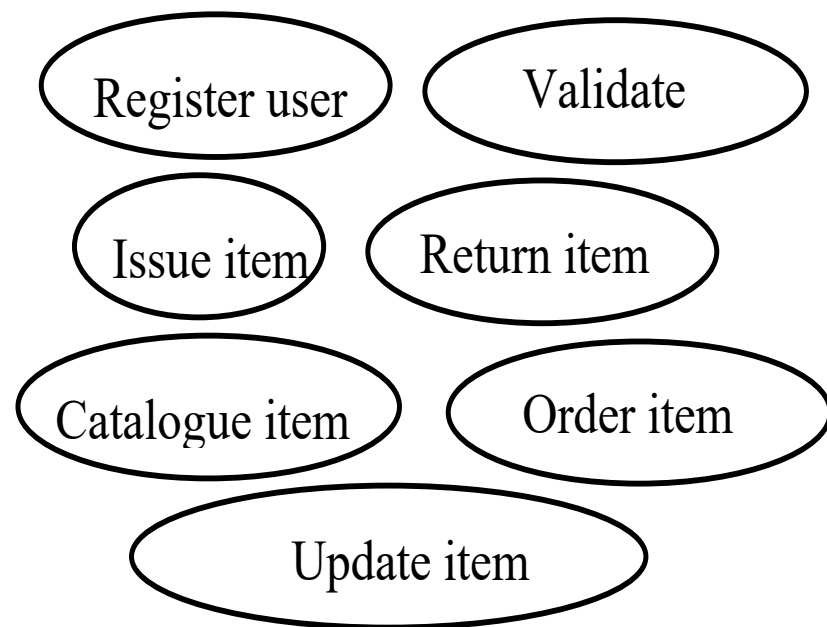
## Пример: Гранични и определящи гледни точки

---

### *Bounding Viewpoints*



### *Defining Viewpoints*



## Стъпка 2 – Структуриране на гледните точки

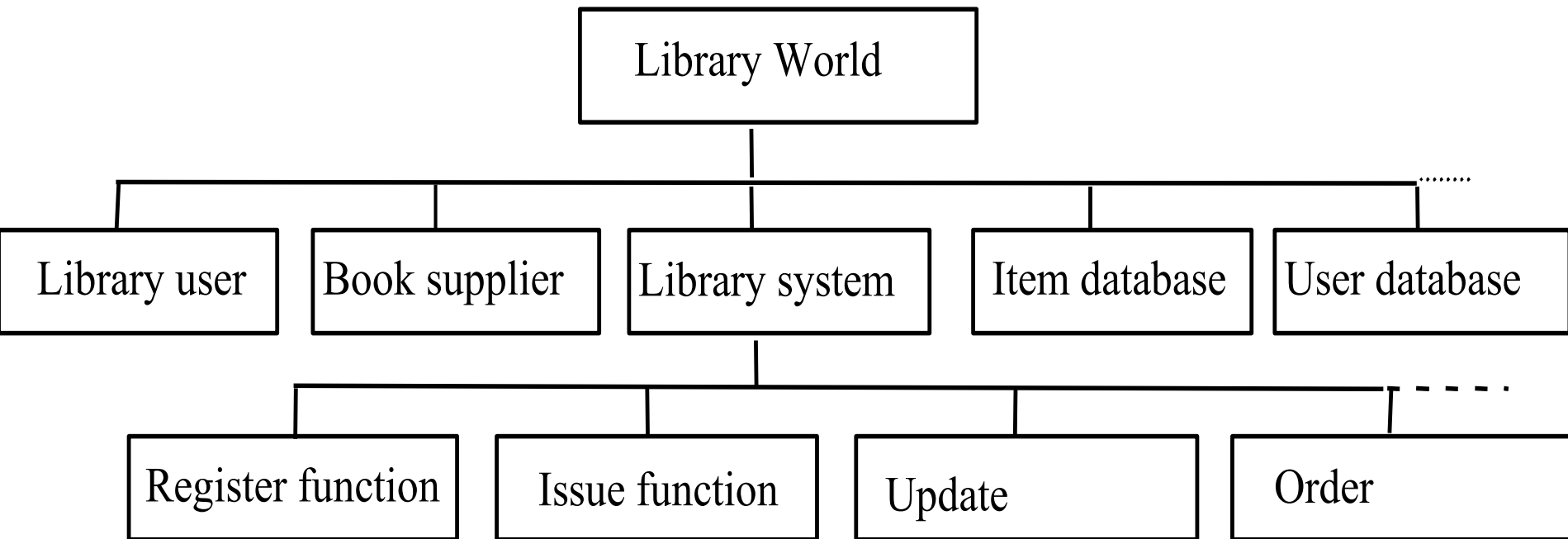
---

Рамка за записване и за анализиране на гледните точки.

- *Итеративна (top-down) декомпозиция на системата в йерархия от функционални подсистеми*
- Структурирано представяне на *bounding* гледни точки - поставят на *същото* ниво, на което е целевата система.
- *Всяка функционална подсистема представлява отделна гледна точка*

# *Пр. Библиотечна система – структуриране на гледните точки*

---



# Стъпка 3 – Таблична сбирка (Tabular collection)

---

- **Механизъм за *събиране* на информация за дадена гледна точка**
- Всяка гледна точка се разглежда последователно по отношение на *действието, което извършва*:

*Данните, използвани за тези действия, получените изходни данни, източникът на данните и дестинацията на данните*
- *Табличните сбирки* имат за цел и да *разкрият пропуските и конфликтите* в потока на информацията между отделните гледни точки и така да **осигурят консистентност**.

# Библиотечна система - tabular collection

Source	Input	Action	Output	Destination
Library user	requested item	check item	issued item	Library user
			error message	Issue clerk
Library user	library card	validate user	loan default message	Issue clerk

## Стъпки 4-7

---

- Стъпката (4) на *структуриране* на данните включва декомпозиция на елементите от данните до *съставните им части* и създаване на *речник на данните*.
- Моделиране на *дейностите на отделните (стъпка 5) и комбинираните (стъпка 6) гледните точки* чрез използване на диаграми на дейностите от вида на DFD - с нотацията на метода SADT.
- Последната стъпка в CORE включва извършване на *анализ на ограниченията* върху системата като цяло.

# Недостатъци на CORE

---

- Понятието за гледна точка е *слабо* дефинирано
- Анализът се съсредоточава върху вътрешните перспективи – defining (определящите) гледни точки  
Bounding (граничните) гледни точки не се анализират повече от това *да се разглеждат като приемници и източници на данни*.
- Заради вградената структура CORE трудно се интегрира с други методи за изискванията.

# Системно инженерство, базирано на гледни точки /Viewpoint-oriented system engineering (VOSE)/

---

- Разработен в Imperial College, Лондон в началото на 90-те
- Гледните точки (in VOSE) *обхващат* ролята и отговорността на даден участник в определен етап от разработката.
- VOSE използва гледните точки, за да *разделя и да разпределя дейностите* и *знанието* на участниците в разработването на софтуера.
- Системното инженерство, базирано на гледните точки, е *рамка за интегриране* на методи за разработка на система.



# Шаблон за гледните точки

---

- Една гледна точка във **VOSE** може да се счита за шаблон, който описва какво вижда участника:
  - *Стил* - представителна схема (начин на представяне) на това, което вижда
    - *Област* (problem domain)
    - *Спецификация* (partial)
    - *Работен план*
    - *Работен запис* (development history)

# Стандартни слотове на шаблон за гледните точки

---

## **Style**

Definition of representation

## **Work Plan**

Development actions and rules

## **Domain**

Problem domain described by  
ViewPoint

## **Specification**

actual partial  
specification

## **Work Record**

Development  
history

# Viewpoint конфигурации

---

- Гледните точки могат да се организират в *конфигурации* като колекции от свързани гледни точки.
- Една конфигурация може да се състои от
  - Шаблони с *различни стилове*, разглеждащи **един и същ дял** на проблемната област, или
  - Шаблони с *един и същ стил*, разглеждащи **различни дялове** на проблемната област
  - *Крайната система е комбинация от конфигурациите*, на които всички конфликти са решени.

# Пример: Библиотека

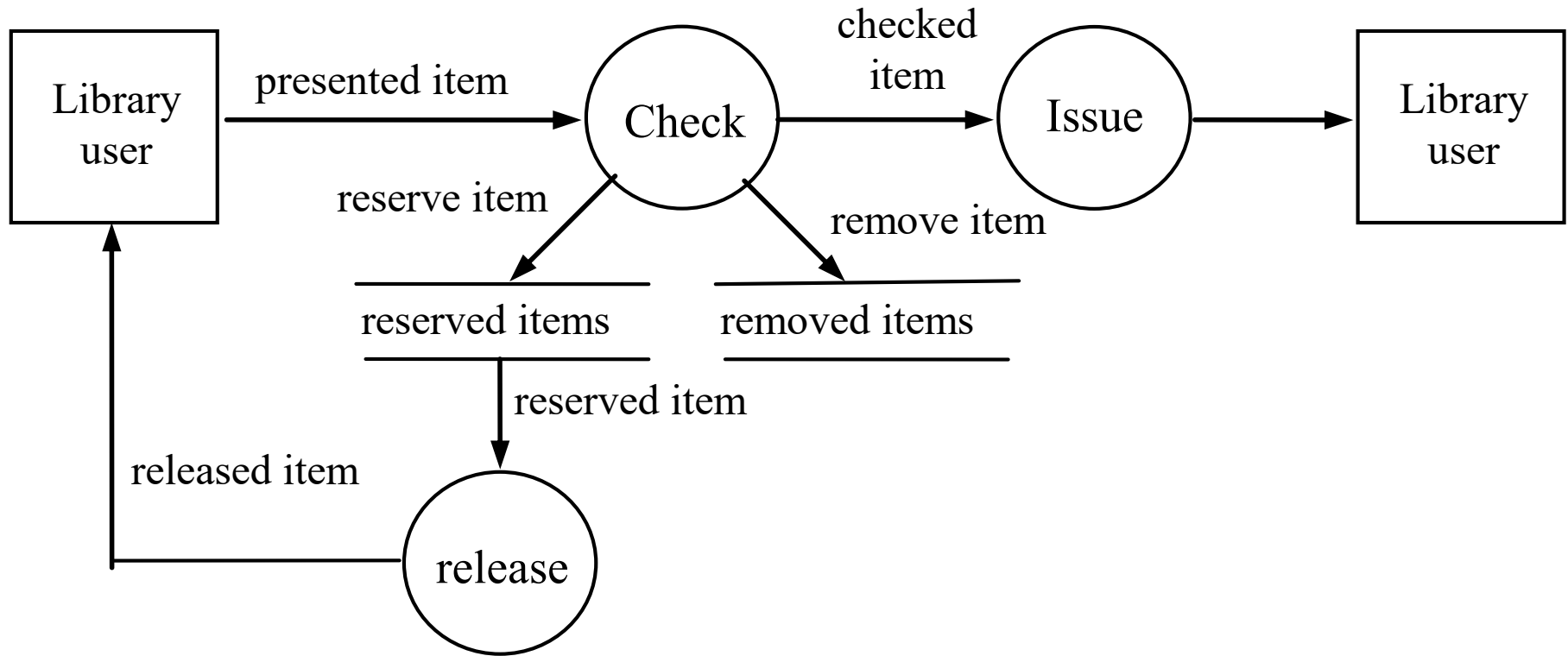
---

Нека имаме артикул от библиотеката, който потребителят представя за *заемане*, *връщане* или *резервиране*.

- „Библиотечният свят“ може да бъде разделен на области (гледни точки) на *issue desk* и на *library user* (читател).
- Използват се модели на *потока на данните* и *прехода на състоянията*, с които се моделира библиотечният артикул от гледна точка на всяка област.

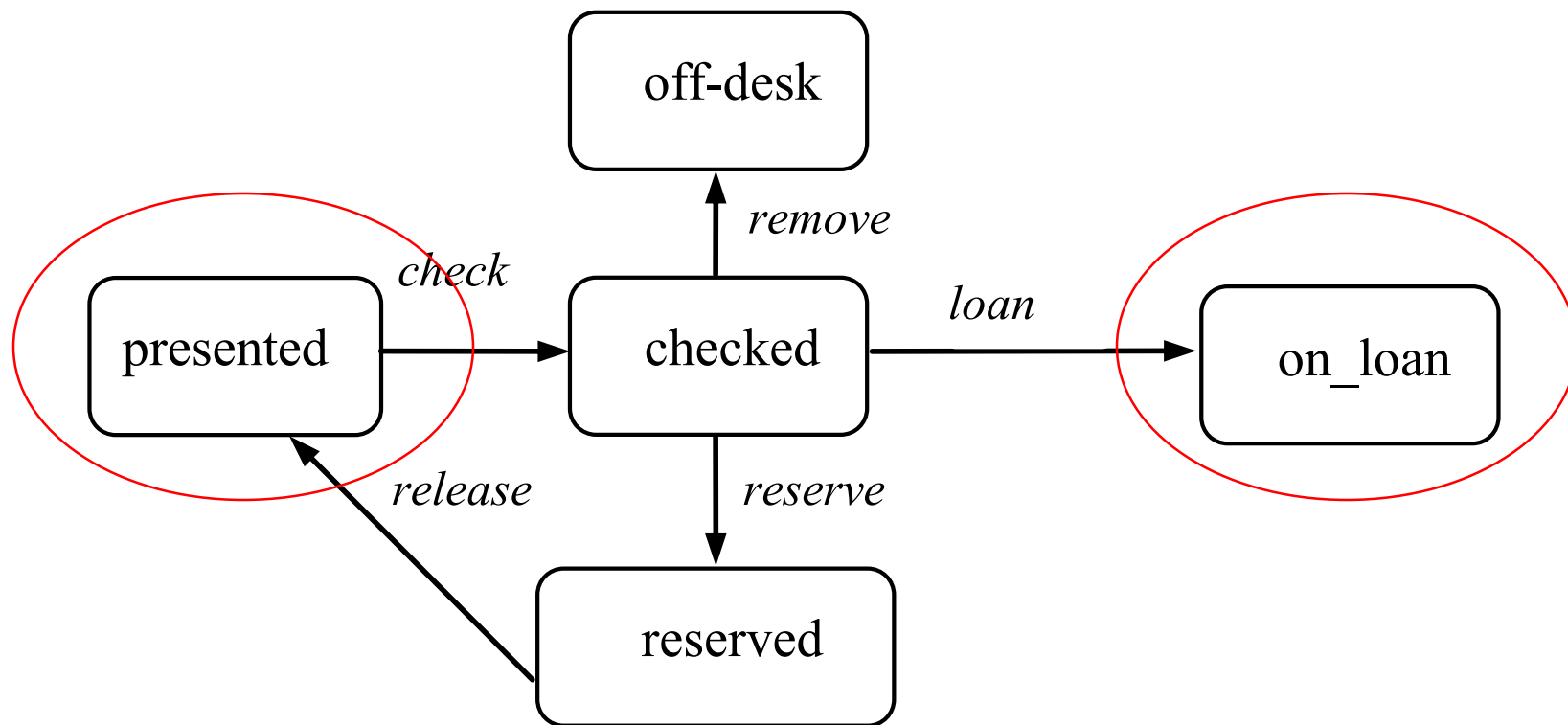
# Модел на потока на данните на **област на Issue desk** с 3 процеса

---



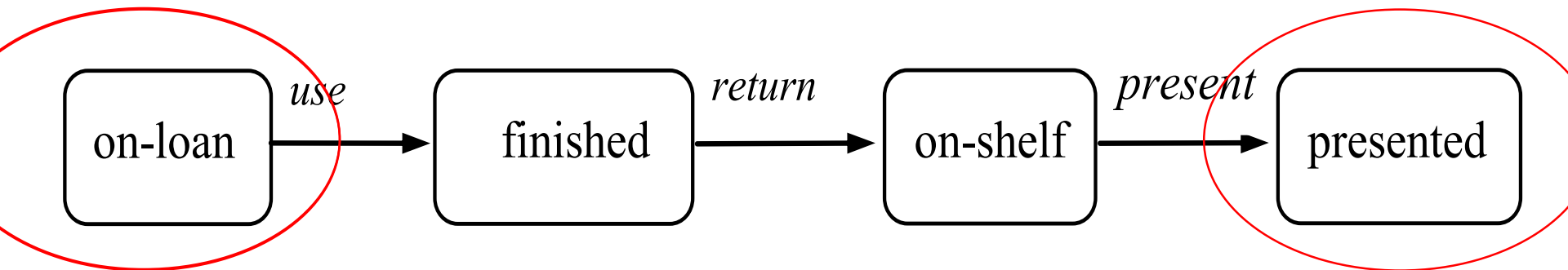
# Модел на прехода на състоянията на **област на Issue Desk**

---



# Модел на състоянията на библиотечната единица от гледна точка на **Library user** **domain**

---



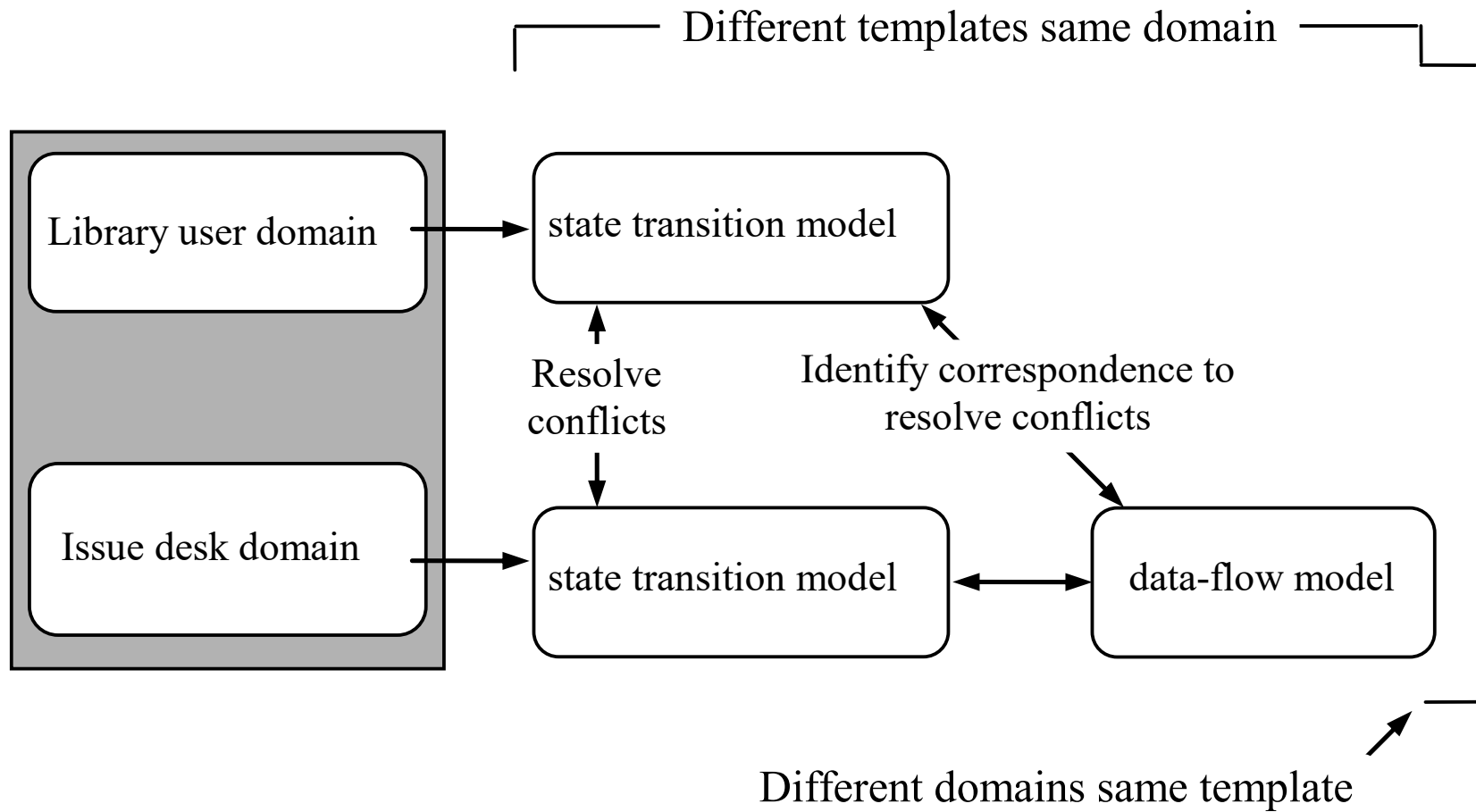
# Свързване на изискванията с използване на подхода на гледните точки (Разрешаване на конфликти)

---

- Важно е, за да се осигури *съгласуваност* между различни представяния на областите
- За еднаквите стилове, виждащи различни домейни, конфликтите се разрешават чрез проверка за *последователност между моделите* т.е. да осигурим последователност в информационните потоци между отделните части на описанията.
- За различните стилове, виждащи еднакви домейни, трябва да се открият *съответствията между схемите за представяне*, за да се улесни проверката за съгласуваност.

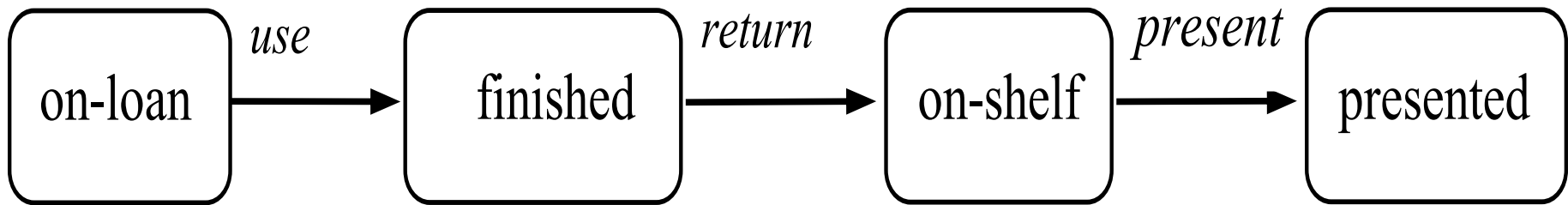


# Проверка за съгласуваност



# Съответствие между преход и функция

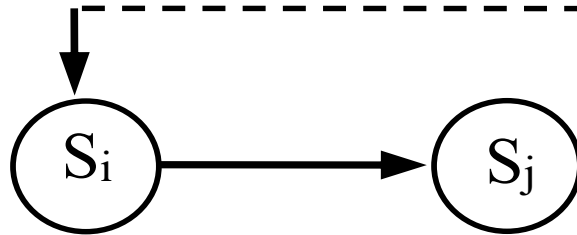
---



# Съответствие между състояние и данни

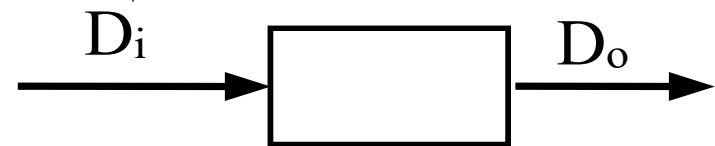
---

*State transition analysis*



State

*Data-flow analysis*



Data-flow

# Mapping different templates, same domains

Issue desk DFD	Issue desk ST
check	check
issue	loan
release	release

# Mapping on different domains, same template

<b>Issue desk ST</b>	<b>Library user ST</b>
presented on-loan	presented on-loan

# Дефиниране на изискванията, базирано на гледни точки (View Point Requirements Definition (VORD))

---

- Разработен в университета в Ланкастър през 90-те години.
- **Използва модел, ориентиран към услуги (*service-oriented*)**.  
Гледните точки са аналог на клиента.
- Главно предназначен за специфициране на *интерактивни системи*
- Базиран на гледни точки, които се фокусират върху *потребителски проблеми и организационни въпроси*
- VORD дефинира **два** главни типа **гледни точки**: *преки и косвени*

# Преки и косвени гледни точки

---

- **Преки гледни точки**

- Взаимодействат директно със системата
- Те съответстват *на клиенти, които получават услуги от системата и предоставят информация*
- Включват *оператори/потребители* или *други подсистеми* с интерфейс към анализираната система

- **Косвени гледни точки**

- *Не взаимодействат директно* със системата
- Те се „интересуват“ от някои или от всички услуги на системата
- Генерират изисквания, които *ограничават услугите*, предоставяни на преките гледни точки
- Включва гледни точки *на организацията, на околната среда, на проектирането и разработването*

## *Примери* за преки и косвени гледни точки

---

- Гледна точка на системно планиране, която засяга бъдещо предоставяне на библиотечни услуги (косвена)
- Потребител на библиотеката, който достъпва до системните услуги чрез интернет (пряка)
- Търговска гл. т., която описва ефекта на представянето на системата сред персонала и задълженията на персонала в библиотеката (косвена)



## Подход за *валидиране* на изискванията, базиран на гледните точки (Leite and Freeman, 1991)

---

- Използва гледни точки, за да подпомага *ранното* валидиране на изискванията.
- Цел на подхода е *да* идентифицира и да класифицира проблемите, свързани със *завършеността и точността* на описанието на системата

# Гледни точки, перспективи и изгледи (views) – дефиниции според метода за валидиране

---

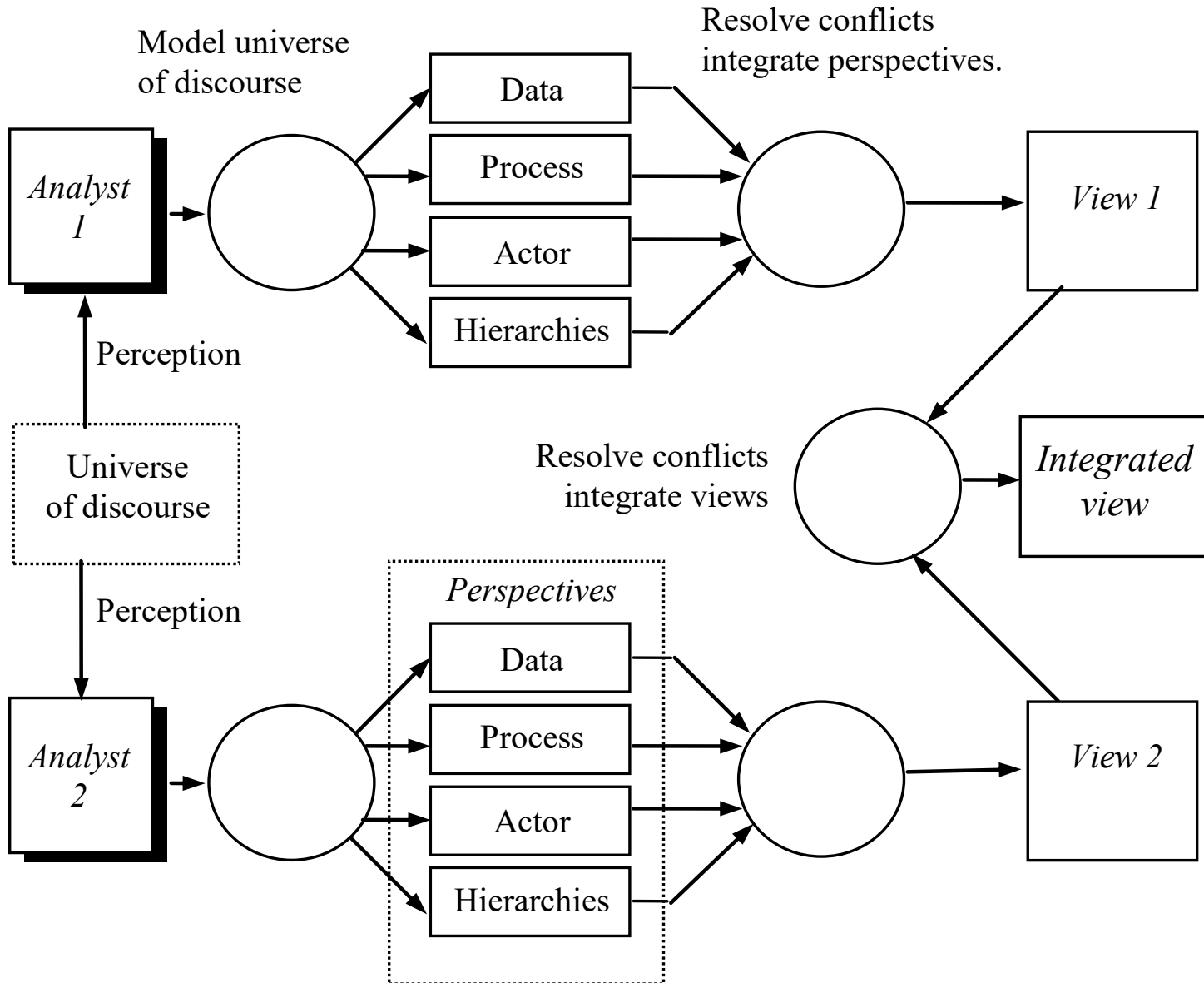
- *Гледна точка* се дефинира като позиция (standing position), която някой човек използва, когато изследва обхвата на проблема/системата
- *Перспектива* се дефинира като набор от факти, които се наблюдават и моделират според определен аспект на реалността
- *Изглед (View)* се дефинира като обединение на перспективите
- За представянето на гледните точки се използва *език на гледните точки (VWPL)*

# Стъпки на метода

---

- Включва *поне двама* аналитици (гледни точки), използващи VWPL
- Перспективите се обединяват в изглед
  - Expressed in the process perspective together with the hierarchies
- Изглед (View) се конструира чрез описание на проблема с използване на три перспективи - *данните, процеси и актьори*
  - Аналитиците използват is-a и part-of йерархии, за да подобрят своя собствен изглед (view)
- Перспективите и йерархиите се анализират и се съставя ‘списък с несъответствия’ и ‘видове несъответствия’
- Когато има поне два изгледа се сравняват отделните гледни точки за точност и пълнота.

# Процес за валидиране на изискванията, базиран на гледните точки



# Ключови моменти

- Анализът на изискванията е разпределен процес, който включва много участници с различни интереси
- Гледната точка е съвкупност от информация за дадена система или съответен проблем, околна среда или област, която е събирателна (collective) от определена разбиране на (обикновено) актьор, заинтересовано лице или външен обект/система.
- Техниките на структурния анализ (SADT) нямат явно дефинирани гледни точки
- CORE има два типа гледни точки – гранични и определящи.
- VOSE е подход, който предоставя шаблон за описание на гл. точка - какво вижда, каква област представя, спецификация, работа и план.
- VORD има преки и косвени гледни точки.