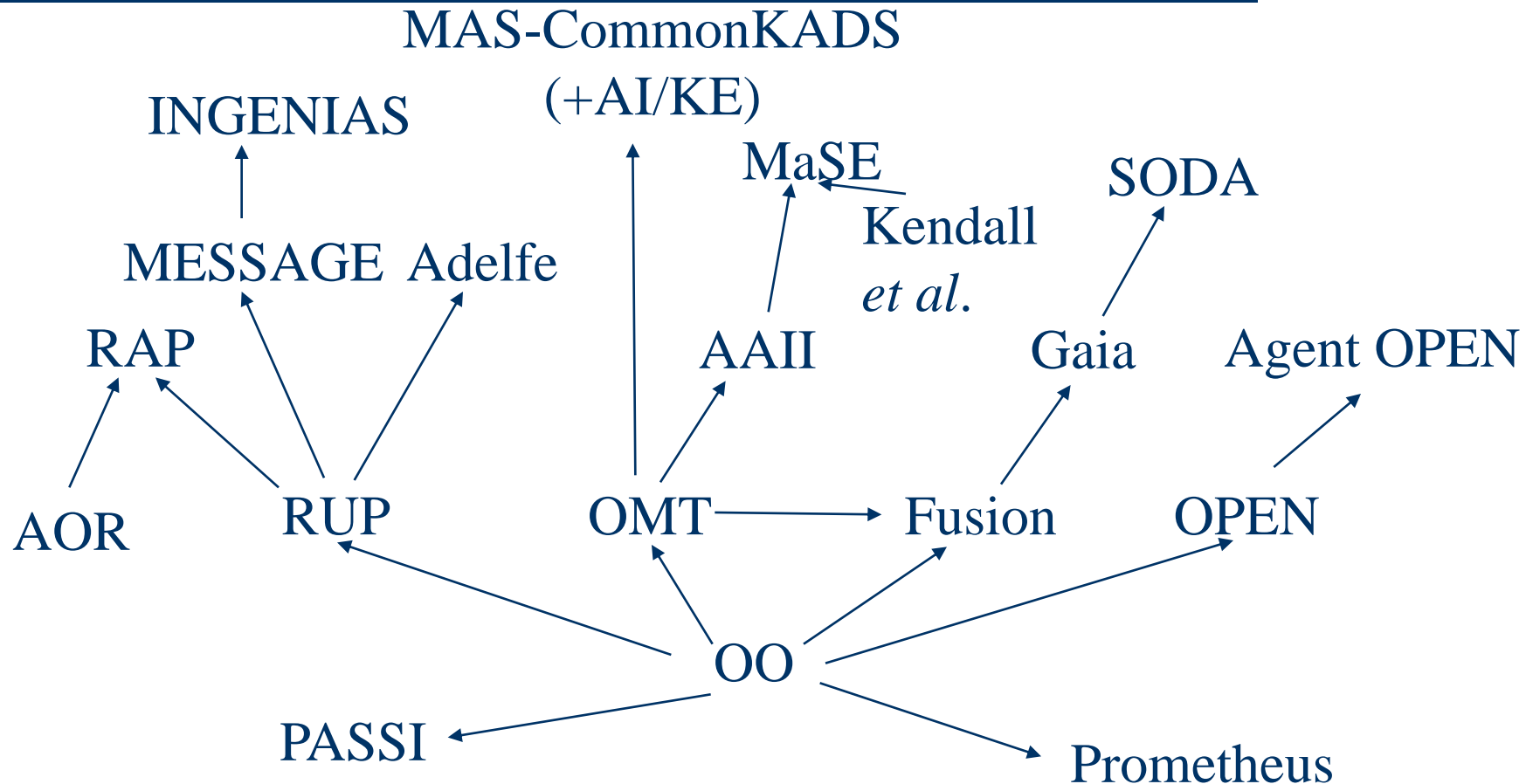


Методологията GAIA: Агент-Ориентиран Анализ и Дизайн

Аделина Алексиева-
Петрова

Хронология



Какво е GAIA методология?

- Софтуерна методология за анализ и дизайн на агент-базирани системи.
- Позволява да се анализират всички изисквания и да се изгради дизайн, които да може да се реализира директно.
- Анализът и дизайнът могат да се разглеждат като процес на разработване на все по-подробни **модели** на конструираната система.

Gaia (1)

- **Gaia** е първоначално предложена от M. Wooldridge, N.R. Jennings, and D. Kinny в статия "A methodology for Agent-Oriented Analysis and Design" (**1999**)
- Нова версия е предложена от M. Wooldridge, N.R. Jennings и Franco Zambonelli (3-10-**2003**).

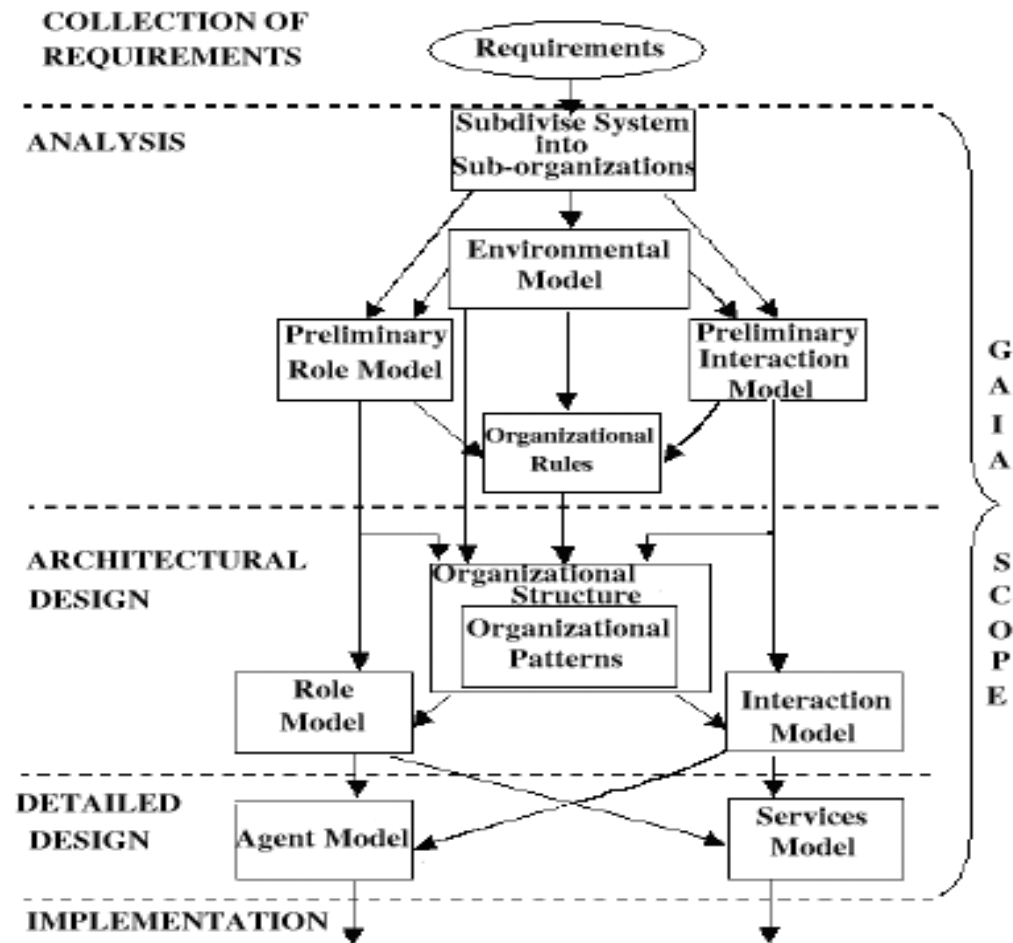


Fig. 6 Article: Developing Multiagent System: The Gaia Methodology

Gaia (2)

- **Изисквания:** Gaia не обхваща действията свързани с изискванията към приложението
- **Анализ:**
 - **Организация:** множество организации трябва да съществуват едновременно в системата и да има автономно взаимодействие MASs (Multi-Agent Systems)
 - **Модел на средата:** е моделирана по отношение на абстрактни изчислителни ресурси, като променливи, предоставена от агенти за наблюдение, за извършване или за консумация

reads *Var1 // readable resource of the environment*

Var2 // another readable resource

changes *Var3 // a variable that can be also changed by the agent*

Gaia (3)

- **Предварителен Модел базиран на ролите:** определянето на ролята на организацията и протоколи. Има два основни атрибут, които се класифицират като:
 - **Разрешения:** какво може или не може да бъде направено, докато са изпълнявали ролята
 - **Отговорности:**
 - Liveness: "something good happens"
 - Safety: "nothing bad happens "
- **Предварителен Модел на Взаимодействие:** дефиниция за протокол за всеки тип взаимодействие между ролите със следните атрибути: има на протокол, инициатор, партньор, входове, изходи, описание.

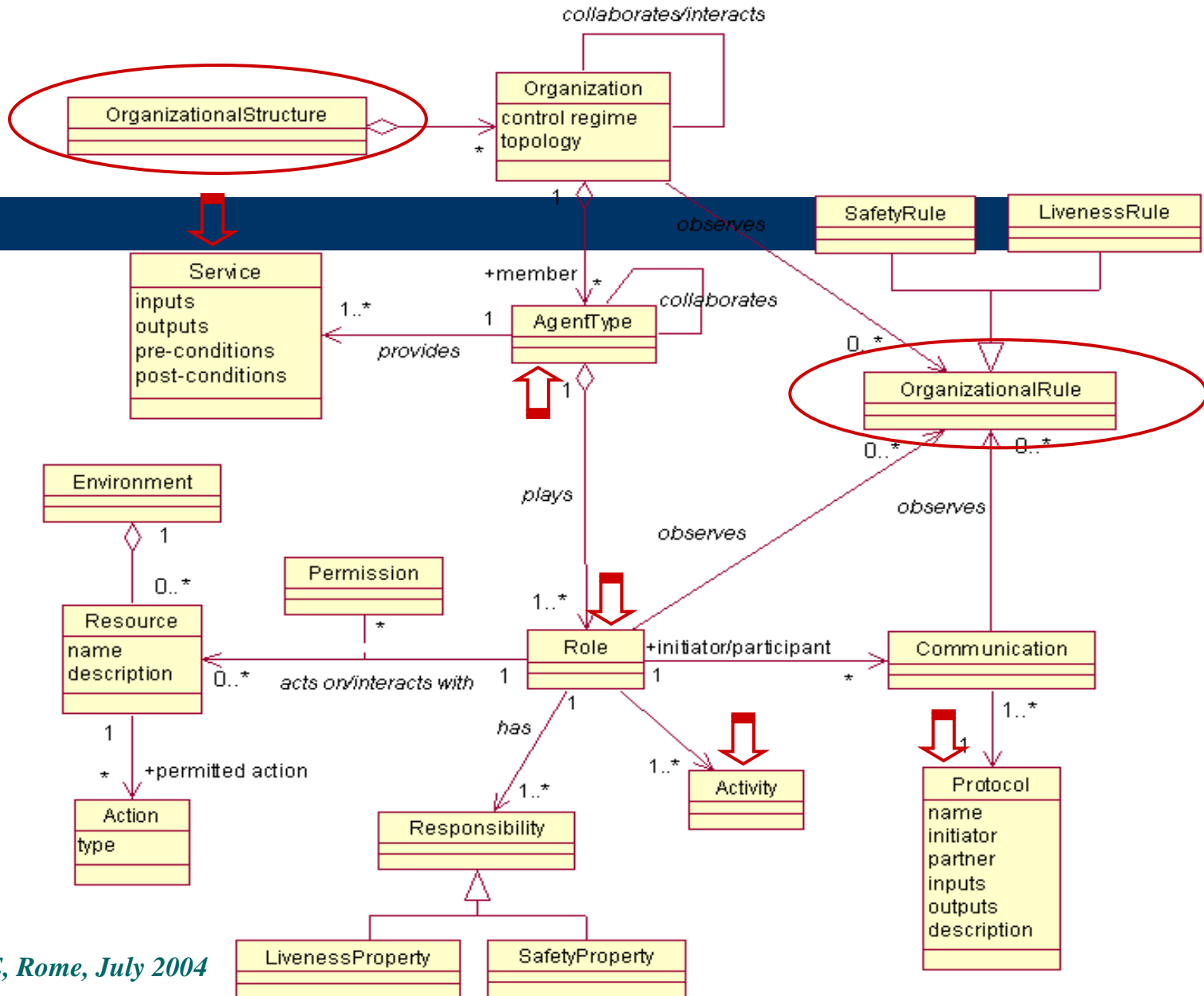
Gaia (4)

- **Организационни правила:** има разлика в зависимост от свойствата liveness и safety при определяне на правилата.
 - Liveness правилата определят как организацията трябва да се развива с течение на времето.
 - Safety правилата дефинира независимо от времето глобалните варианти на организацията.

Gaia (5)

- **Дизайн:**
 - По време на този етап, различни нотации и графичните изображения могат да бъдат приети за описание и представяне на роли и взаимодействията между тях.
- **Детайлен дизайн:**
 - **Модел на агента:** определя класовете агенти които трябва да се дефинират за реализиране на определена роля и колко инстанции на всеки клас ще бъдат създадени.
 - **Модел на услугите:** определя услугите свързани с всеки клас агент. За всяка услуга се документира неговите входи, изходи, предварителни условия и след условия трябва да бъдат предоставени.
- **Имплементация:** Gaia не засяга тази фаза

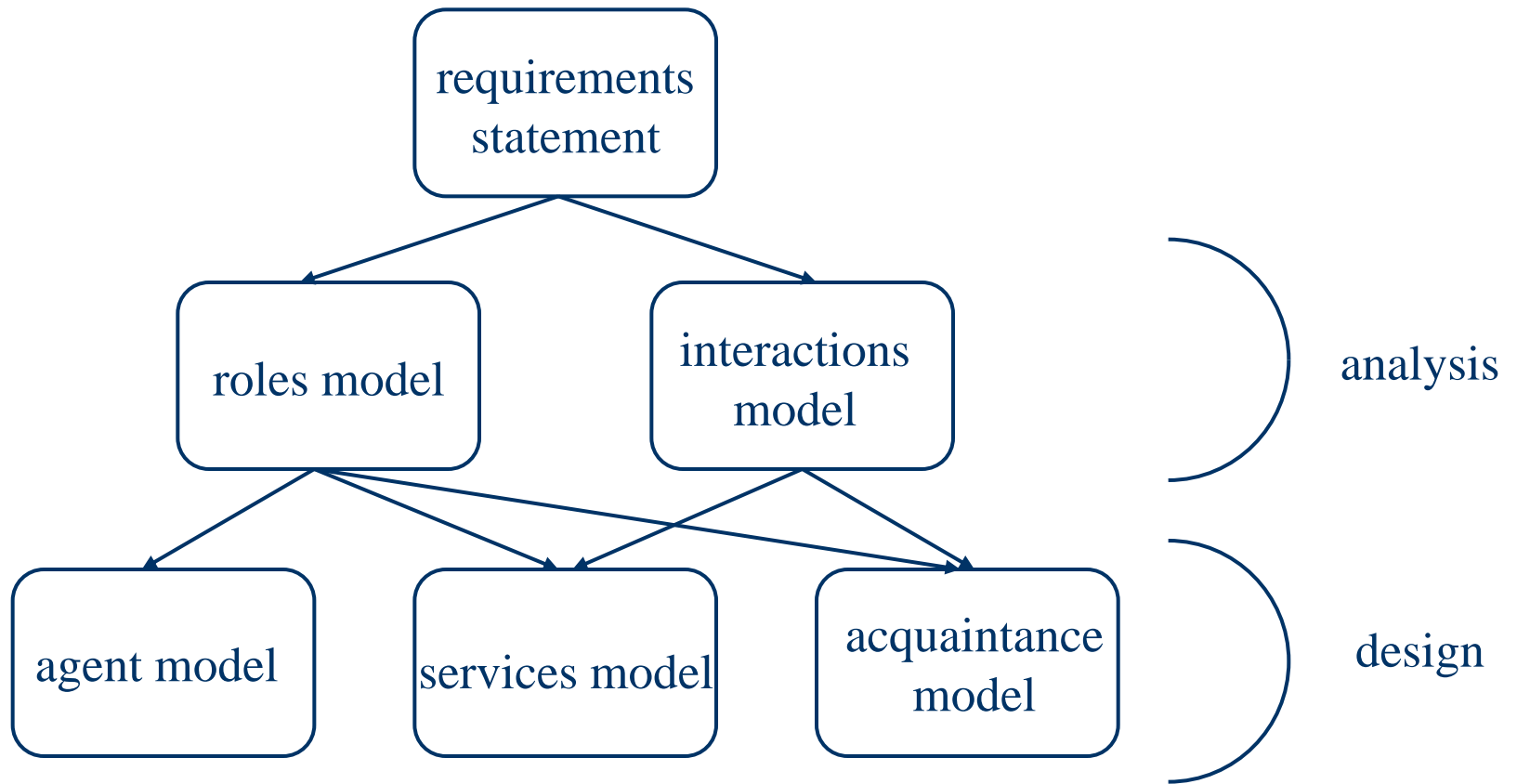
Метамодел на Gaia



Методология (1/2)

- Използвайки GAIA анализът преминава от абстрактни към конкретни концепции.
- Анализът и дизайнът могат да се възприемат като процеси на разработване на детайлните модели в системата.

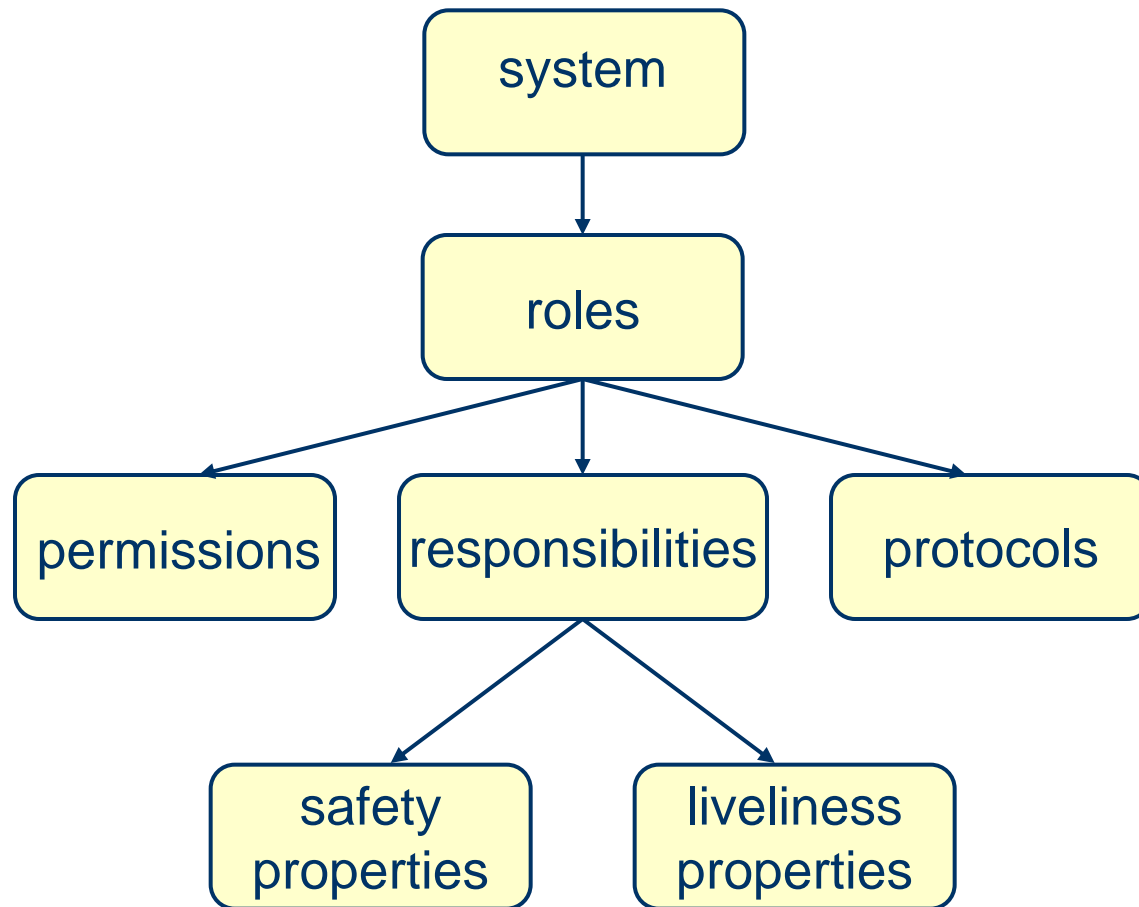
Методология (2/2)



Анализ

- Предмет на анализ модела е да създаде разбираемост в системата и нейната структура.
- Това се постига, чрез организацията на системата.
- Организацията се възприема като колекция от роли.

Методология



Основни концепции (1/2):

- Система: т.е. общество от агенти
- Роли
 - Например: президент, секретар
 - В действителна организация това се действителни хора/ агенти.
 - Агентът може да изпълнява няколко роли, същата роля може да се поемат от множество агенти.

Основни концепции (2/2):

- Ролята се описва от 4 атрибута:
 - Отговорности: liveness свойства + safety свойства
 - Права: == ресурси
 - Действия
 - Протоколи: например: Dutch auction, English auction и др.

Модел базиран на ролята:

- Идентифицира ключовите роли в системата.
- Тук ролята може да се разгледа като абстрактно описание на единици с очаквани функции.

Отговорности (responsibilities)

- Всяка роля се създава за да извършва дейност. Тя има известна функционалност, която е представена от атрибутите, известни като отговорности на ролята.
- Функционалността на ролята се определя от нейните отговорности. Те могат да се разделят в 2 категории: Liveness и Safety .

Liveness (1/2)

- Liveness отговорности са тези които определят, че „нещо добро ще се случи“. Те имат положителен смисъл и постановява, че агента, изпълняващ ролята е още жив.

Liveness (2/2)

- Liveness характеристики са специфицирани чрез liveness изрази, които дефинират жизнения цикъл на ролята.
- Liveness изразите определят потенциалните траектории на изпълнение на различни дейности и взаимодействия свързани с ролята. Форма на liveness израза е:

RoleName=*expression*

Пример

- CoffeeFiller can
 - Take water from the supply
 - Take electricity to heat the water
 - Take coffee from CoffeeStock is not empty

Пример

CoffeeFiller = (All)^ω

All = Fill.InformWorkers.CheckStock.AwaitEmpty

Оператори за liveness израз

$x.y$	x followed by y
$x y$	x or y occurs
x^*	x occurs 0 or more times
x^+	x occurs 1+ times
x^ω	x occurs infinitely often
$[x]$	x is optional
$x y$	x and y interleaved

Safety Свойства

- Изготвя се списък на безопасните изрази, където всеки елемент от списъка изразява индивидуални отговорности за безопасност.
- Предполага се, че тези отговорности се прилагат във всички състояния на системата при изпълнение.
- Safety пример

$\text{coffeeStock} \geq 0$ Ensures that the coffee
stock never is empty

Права

- За реализиране на отговорностите, ролята има набор от разрешения.
- Това са правата принадлежащи на ролята.
- Разрешенията на ролята идентифицират ресурсите, които са налични за конкретната роля.

Права

- В системите, които обикновено се моделират, разрешенията са информационни ресурси.
- Например, ролята може да се асоциира с възможността да се чете информация, или да се модифицира друга част от информацията.
- Ролята също така трябва да има възможността да генерира информация.

Дейности

- Дейностите на ролята са изчисления асоциирани с ролята, която би могла да се изпълни от агента без взаимодействие с други агенти.
- В обобщение, ролята е също идентифицирана с определен брой протоколи, които дефинират начина, по който тя си взаимодейства с други роли.

Схема на Модела на Ролята

ROLE SCHEMA:	<i>name of role</i>
DESCRIPTION:	<i>short English description of the role</i>
PROTOCOLS:	<i>protocols in which the role plays a part</i>
PERMISSIONS:	<i>“rights” associated with the role</i>
RESPONSIBILITIES	
LIVELINESS:	<i>liveliness responsibilities</i>
SAFETY:	<i>safety responsibilities</i>

Пример

ROLE SCHEMA: CoffeeFiller

DESCRIPTION:

This role involves ensuring the coffee is kept filled and informing the workers when fresh coffee has been brewed.

PROTOCOLS:

Fill, InformWorkers, CheckStock, AwaitEmpty

PERMISSIONS:

reads	supplied coffeeMaker	<i>//name of coffee maker</i>
	coffeeStatus	<i>//full or empty</i>
changes	coffeeStock	<i>//stock level of coffee</i>

RESPONSIBILITIES

LIVELINESS:

CoffeeFiller = (Fill.InformWorkers.CheckStock.AwaitEmpty)^ω

SAFETY:

- $\text{coffeeStock} \geq 0$

Модел на Взаимодействие

- Има различни зависимости и взаимоотношения между роли в много-агентни организация.
- В Gaia, тези връзки са представени във модел на взаимодействието .
- Този модел се състои от набор от протоколи, определени по един за всеки тип взаимодействие между роля.

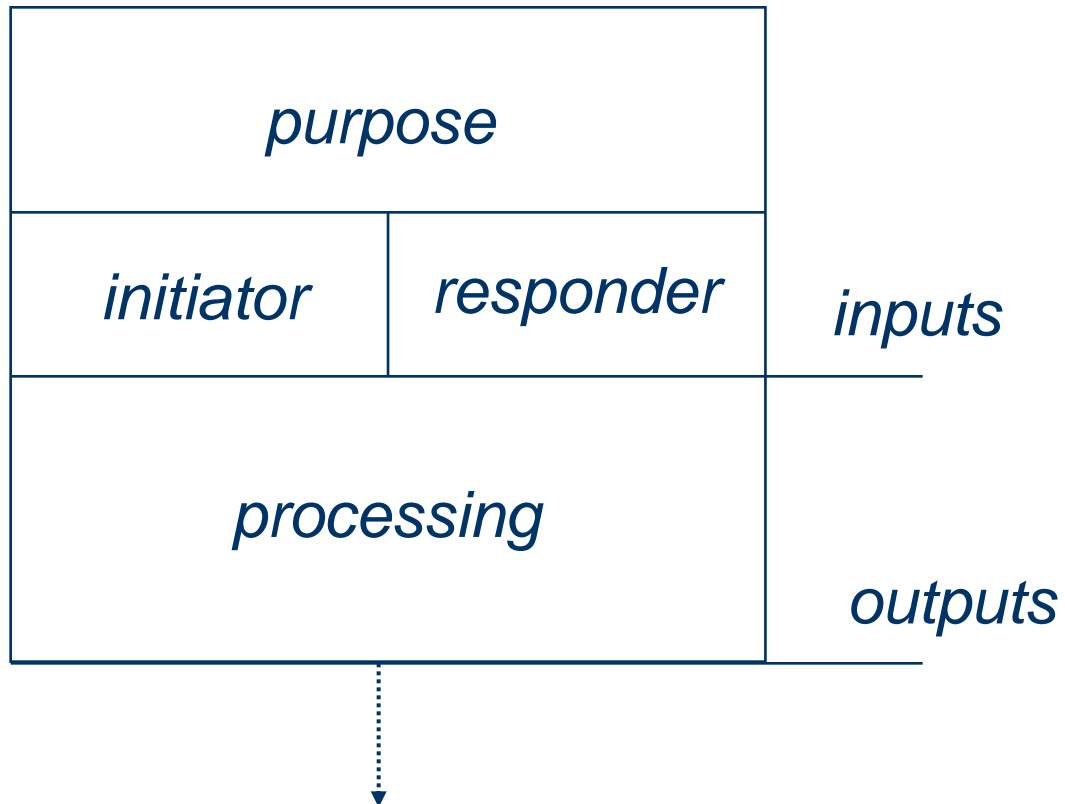
Протокол (1/2)

- Дефиницията на протокола се състои от следните атрибути:
- Цел: кратък текст описващ произхода на итерацията.
- Инициатор: кои е започнал ролята.
- Отговорник: ролите с които инициатора ще взаимодейства.

Протокол (2/2)

- Вход: информация, предоставена от инициатора при влизане в сила на протокола.
- Резултат: информация, предоставена от / към протокола по време на взаимодействие.
- Процес: кратко описание на всички процеси които се случват по време на изпълнение на протокола.

Шаблон на протокол



Пример

TenderContract

QM	CV
select which CV to award contract to	

vettingRequirements

VettingRequest

QM	CV
ask for vetting of customer	

customerDetails

CV	QM
perform vetting and return credit rating	

customerDetails
ratingInfo

creditRating

Definition of
Protocol VetCustomer:

Процес на анализ

1. Определяне на ролите на системата
 - Изход: прототип на модел на ролите
2. Определяне и документиране на протоколите
 - Изход: модел на взаимодействията
3. Разработване на модел на ролите на база модел на взаимодействията
 - Изход: пълен модел на ролите
4. Повтаряне на стъпките от 1-3.

Дизайн

- Цел: превръщане на модела от анализа в модел от ниско ниво, който да може да се реализира програмно.
- Занимава се с въпросите какво се изисква за всеки агент и как агента реализира тези услуги.
Генерират се три модела:
 - Модел на агента
 - Модел на услуги
 - Познавателен модел

Модел на агент

- Дърво на типове агент в което главния възел отговаря на ролите, а другите възли на съответния тип агент.



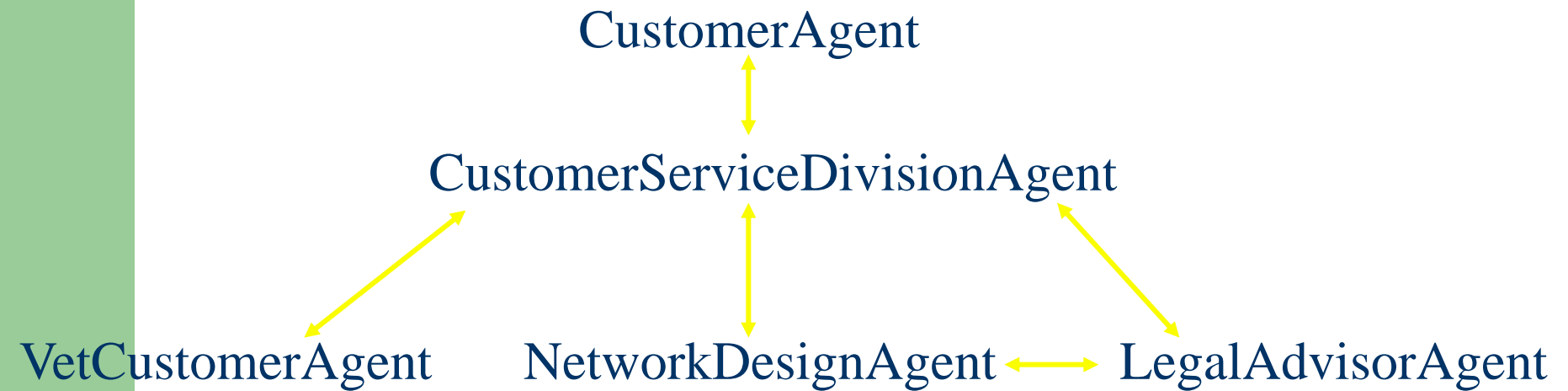
Модел на услуги

- Определя *услугите* свързана с всяка роля на агента.
- Определя основните свойства на тези услуги.
- Определя входа, резултатите, първоначалните условия и пост условията за всяка една услуга.
 - Входа и резултатите се получават от модела на протоколите.
 - Първоначалните условия и пост условията се получават от safety свойствата.

Познавателен модел

- Дефинират се комуникационните връзки между видовете агенти.
- Не се дефинира съобщенията или какво те изпращат.
- Посочват се възможните затруднения
- Състои се от граф между агентите

Пример



Gaia Summary

- Key Concepts
 - Models a MAS as a set of roles with permissions, responsibilities, and protocols
- Analysis
 - Roles Model
 - Interaction Model
 - Analysis Process
- Design
 - Agents Model
 - Services Model
 - Acquaintance Model
 - Design Process