

# O-MaSE

## Organization-based Multiagent System Engineering

MiASI2, TWO2, 2018-2019

# Materiały

- Strona poświęcona metodzie O-MaSE  
<http://macr.cis.ksu.edu/projects/omase.html>  
(Multiagent & Cooperative Reasoning Laboratory, Kansas State University)
- Wybrane publikacje
  - S. DeLoach, J. C. Garcia-Ojeda: The O-MaSE Methodology. [W] M. Cossentino *et al.* (red.), *Handbook on Agent-Oriented Design Processes*, Springer, 2014, 253-285  
[https://link.springer.com/chapter/10.1007/978-3-642-39975-6\\_9](https://link.springer.com/chapter/10.1007/978-3-642-39975-6_9)
  - S. DeLoach: O-MaSE: An Extensible Methodology for Multi-agent Systems. [W] O. Shehory, A. Sturm (red.), *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks*. Springer, 2014, 173-191.  
[https://link.springer.com/chapter/10.1007/978-3-642-54432-3\\_9](https://link.springer.com/chapter/10.1007/978-3-642-54432-3_9)

# O-MaSE

Meta-metoda do opracowywania procesów analizy i projektowania ( $\rightarrow$  *engineering*) systemów wieloagentowych

1. Meta-model definiujący podstawowe pojęcia
2. Zestaw podstawowych kroków procesu (zadań, fragmentów)
3. Zestaw reguł pozwalających na kompozycję wybranych kroków w spójny proces (np. określających wymagania wstępne dla kroków)



1. Opracowanie procesu analizy i projektowania specyficznego systemu wieloagentowego
2. Realizacja opracowanego procesu w celu analizy i zaprojektowania tego systemu

# O-MaSE: Meta-model

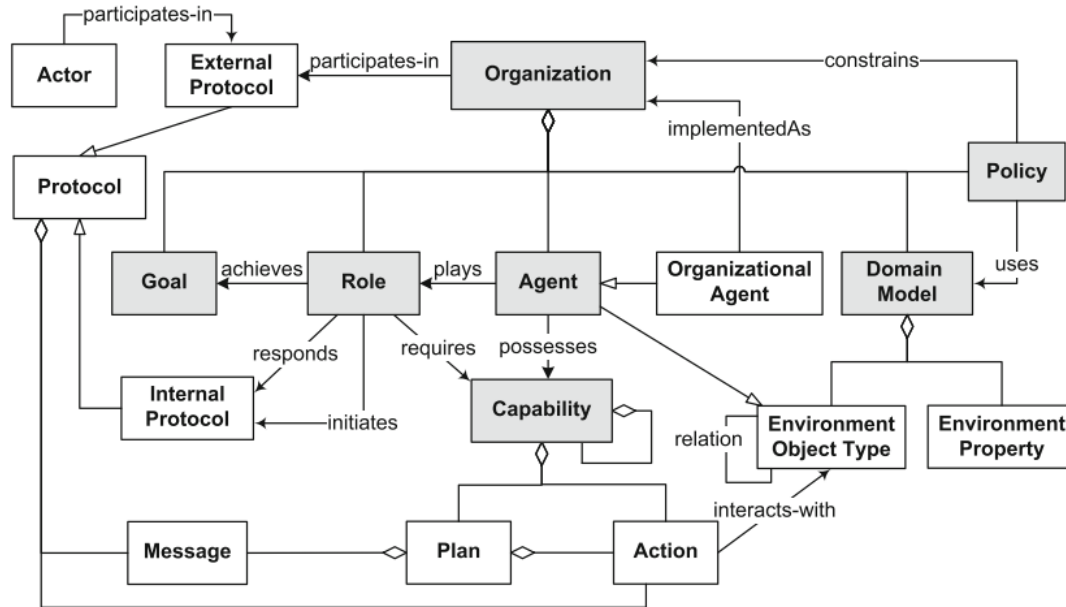


Fig. 9.1 O-MaSE metamodel (multiplicities are omitted for clarity)

# O-MaSE: Podstawowe kroki

**Table 9.1** O-MaSE method fragments

Activities	Tasks	Work Products Created/Modified	Responsible Method-Roles
Requirements Gathering	Requirements Specification	Requirements Spec	Requirements Engineer
Problem Analysis	Model Goals	Goal Model	Goal Modeler
	Refine Goals		
	Model Domain	Domain Model	Domain Modeler
Solution Analysis	Model Organization Interfaces	Organization Model	Organization Modeler
	Model Roles	Role Model	Role Modeler
	Define Roles	Role Description Document	
	Define Role Goals	Role Goal Model	
Architecture Design	Model Agent Classes	Agent Class Model	Agent Class Modeler
	Model Protocols	Protocol Model	Protocol Modeler
	Model Policies	Policy Model	Policy Modeler
Low Level Design	Model Plans	Agent Plan Model	Plan Modeler
	Model Capabilities	Capabilities Model	Capabilities Modeler
	Model Actions	Action Model	Action Modeler
Code Generation	Generate Code	Source code	Programmer

# O-MaSE: Reguły kompozycji

**Table 9.2** Method construction guidelines

Task	Pre-condition	Post-condition
Requirements Specification	True	Requirements Spec
Model Goals	Requirements Spec $\vee$ ((Goal Model $\vee$ GMoDS) $\wedge$ Role Model)	Goal Model
Refine Goals	Goal Model	GMoDS
Model Domain	Requirements Spec	Domain Model
Model Organization Interfaces	Requirements Spec $\wedge$ GMoDS	Organization Model
Model Roles	GMoDS $\wedge$ Organization Model	Role Model
Define Roles	Role Model	Role Description
Model Agent Classes	GMoDS $\vee$ Role Model $\vee$ Organization Model	Agent Class Model
Model Protocols	Role Model $\vee$ Agent Class Model	Protocol Model
Model Policies	GMoDS $\vee$ Organization Model $\vee$ Role Description $\vee$ Agent Class Model	Policy Model
Model Plans	(GMoDS $\wedge$ Role Model) $\vee$ (GMoDS $\wedge$ Agent Class Model)	Plan Model
Model Capabilities	(Role Model $\wedge$ Agent Class Model) $\vee$ Domain Model	Capability Model
Model Actions	Capability Model $\wedge$ Domain Model	Action Model
Code Generation	(Plan Model $\vee$ Protocol Model) $\wedge$ (Capability Model $\vee$ Action Model)	Source Code

# Założenia O-MaSE (1)

- System traktowany jest jako **organizacja agentów**, w której mogą obowiązywać pewne **polityki**
- Agenci współpracują, aby osiągnąć **główny cel** związany z organizacją
- Agenci pełnią określone **role** w organizacji, aby realizować **specyficzne cele** (→ dekompozycja głównego celu)
- Agenci działają zgodnie z zdefiniowanymi **planami**, aby wypełniać swoje role i zrealizować przydzielone cele
- Agenci komunikują się zgodnie z precyzyjnie zdefiniowanymi **protokołami** (→ sekwencja wymienianych wiadomości)
- Agenci komunikują się za pomocą **wspólnego języka** (modelu dziedzinowego) opisanego za pomocą **ontologii**

## Założenia O-MaSE (2)

- **Stacyjny** albo **dynamiczny** przydział ról do agentów
  - Role przypisane do agentów podczas projektowania systemu
  - Role przydzielane agentom podczas działania systemu

- W przypadku dynamicznego przydziału ról

Systemy adaptacyjne

- Role określają wymagania odnośnie pewnych zdolności
- Agenci mają pewne zdolności (w zróżnicowanym stopniu)
- „Rekonfiguracja” systemu w trakcie działania

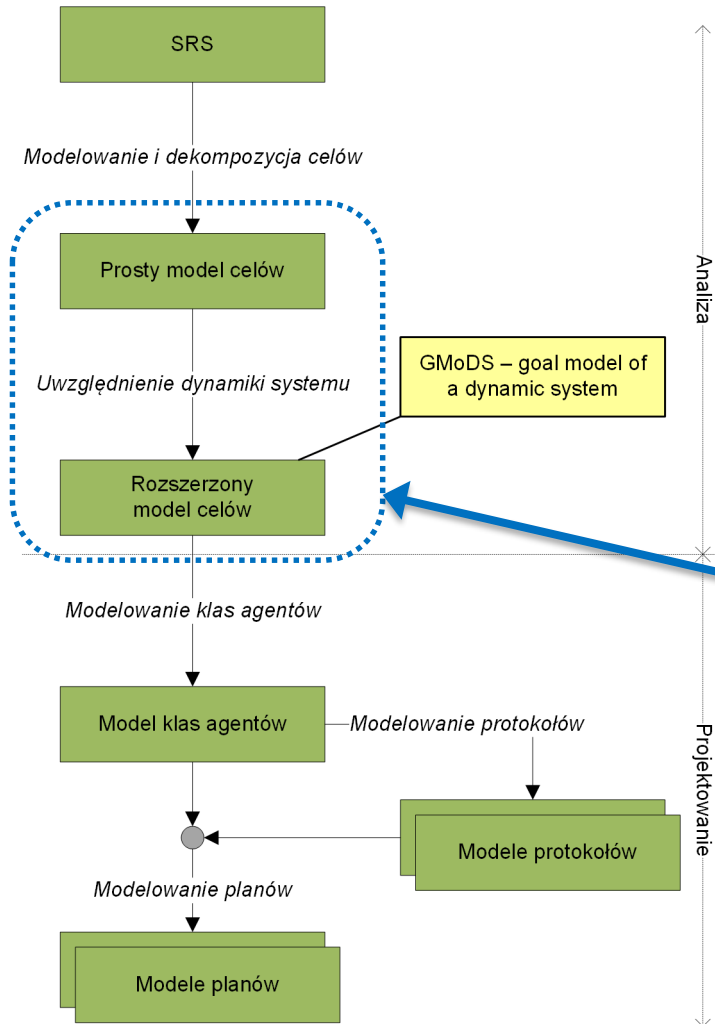
➔ Możliwość rezygnacji z ról – przypisane specyficznych celów bezpośrednio do agentów



# Przykłady procesów dla O-MaSE (1)

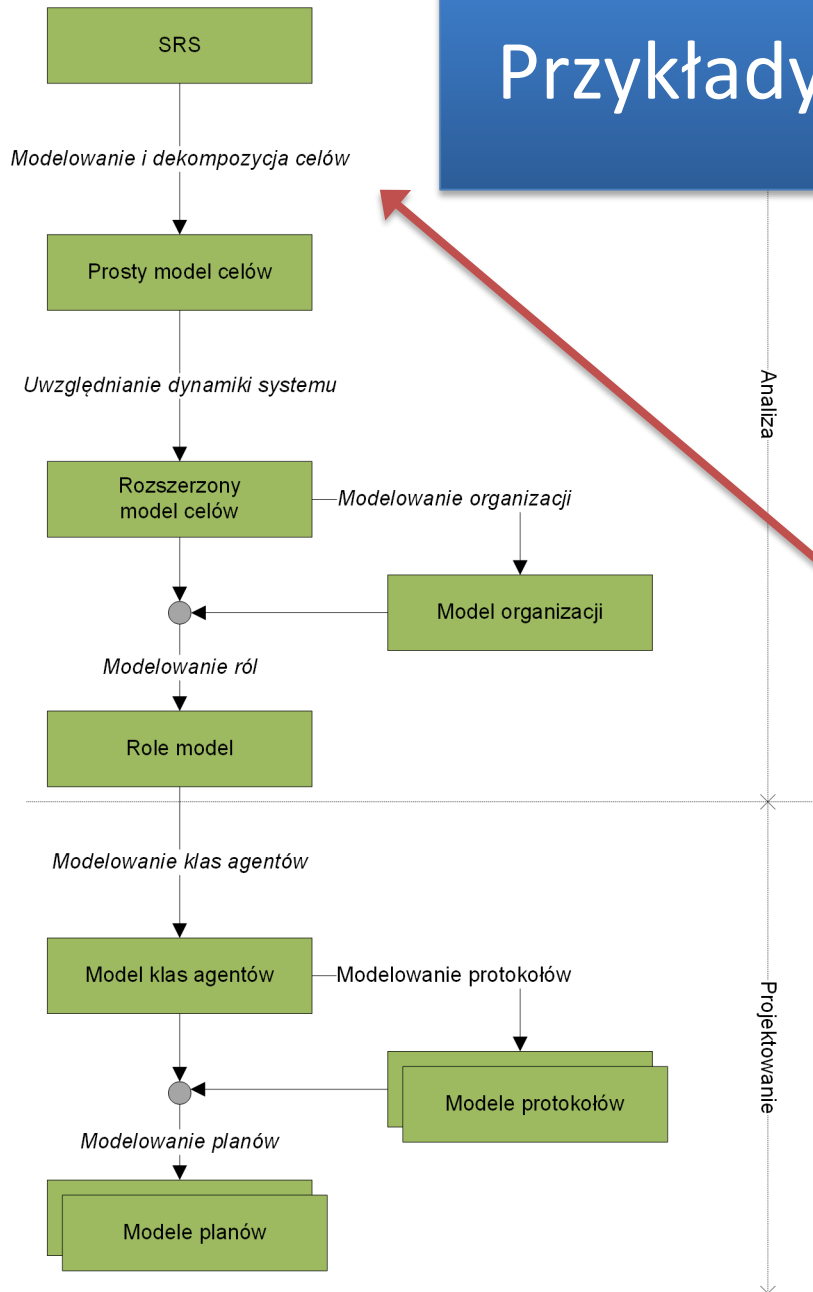
Prosty proces O-MaSE

- brak modelu dziedzicznego
- brak ról (cele przypisane agentom)



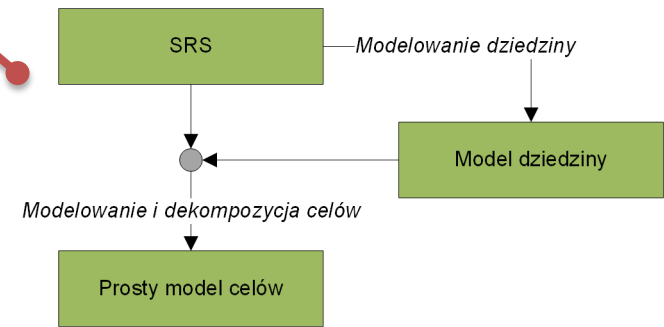
Zazwyczaj opracowanie rozszerzonego modelu celów (GMoDS) w jednym kroku

# Przykłady procesów dla O-MaSE (2)

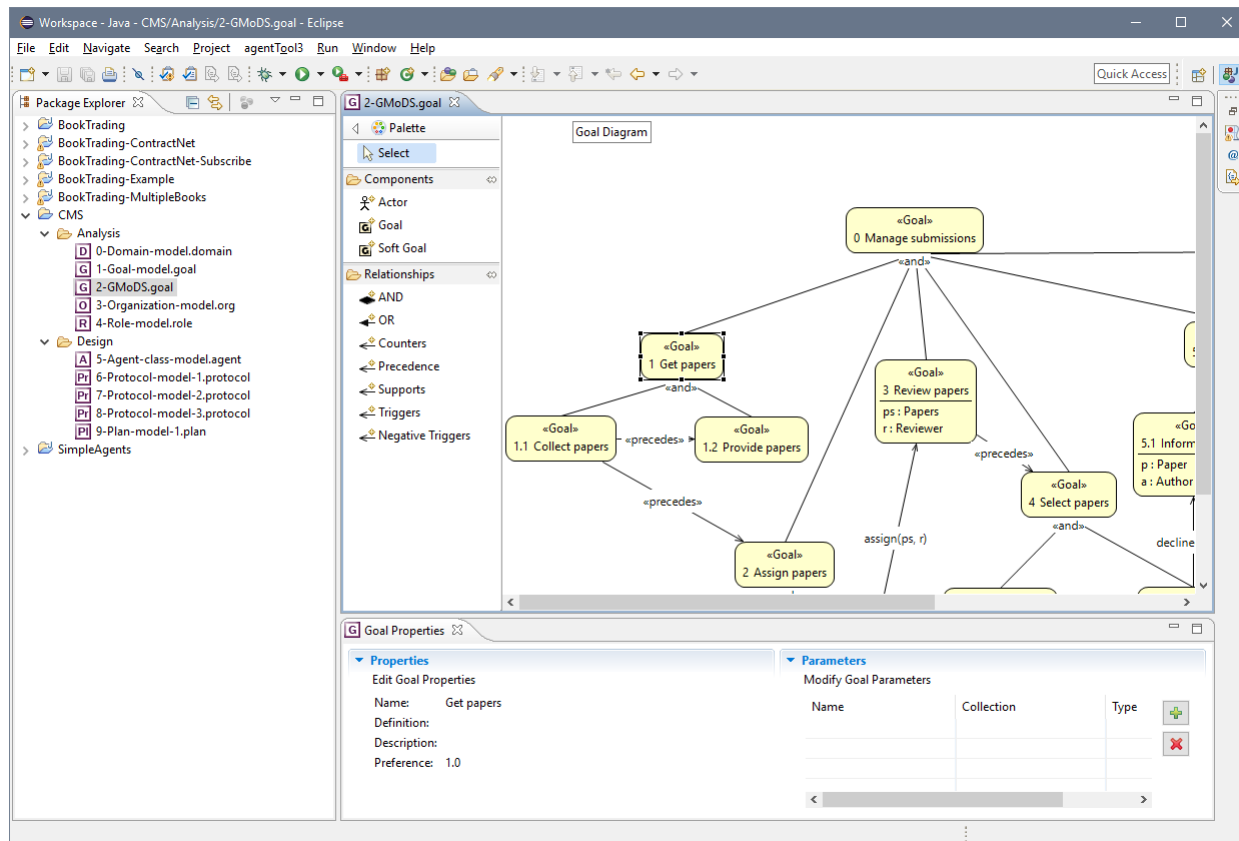


Złożony proces O-MaSE

- role oraz model organizacji
- możliwy model dziedzinowy

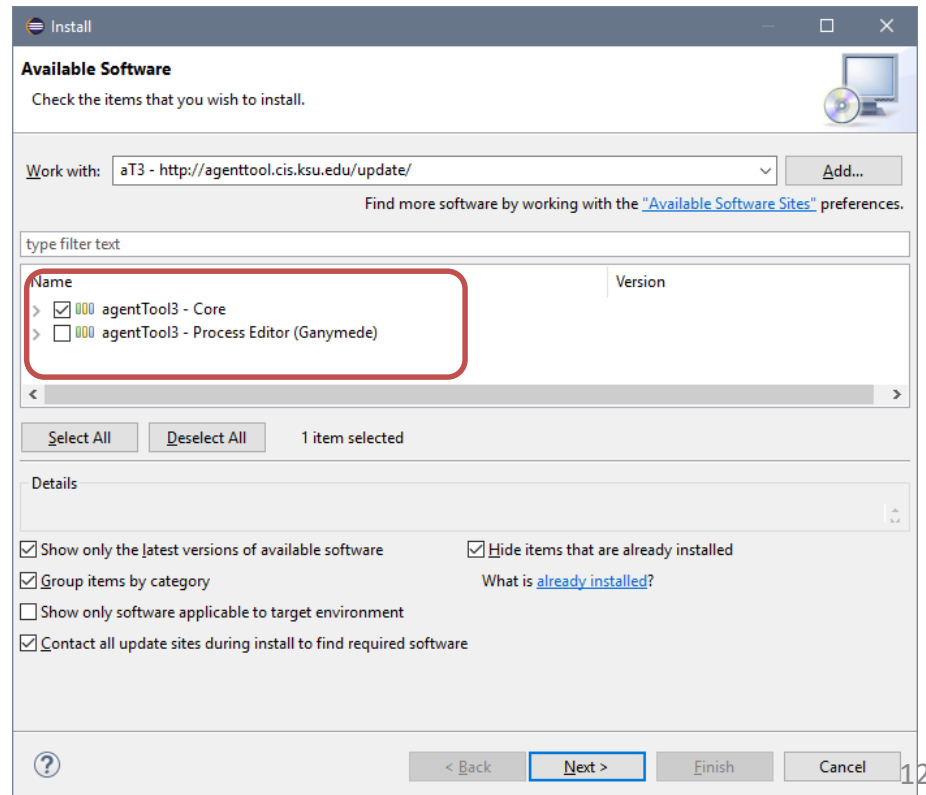
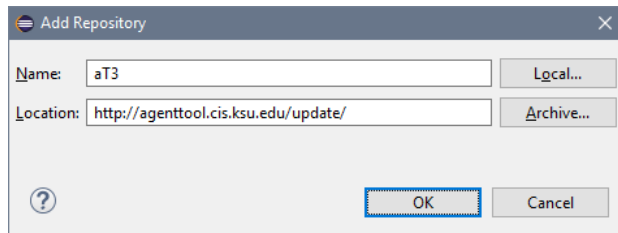


Dodatek do Eclipse pozwalający na tworzenie poszczególnych modeli w ramach O-MaSE – <http://agenttool.cis.ksu.edu/>



# aT3 – Instalacja

1. Help → Install New Software → Add
2. Set Name: aT3, Set Location: <http://agenttool.cis.ksu.edu/update/>
3. agentTool3 - Core



# Przykład – system do zarządzania konferencją (conference management system, CMS)

## Specyfikacja wymagań

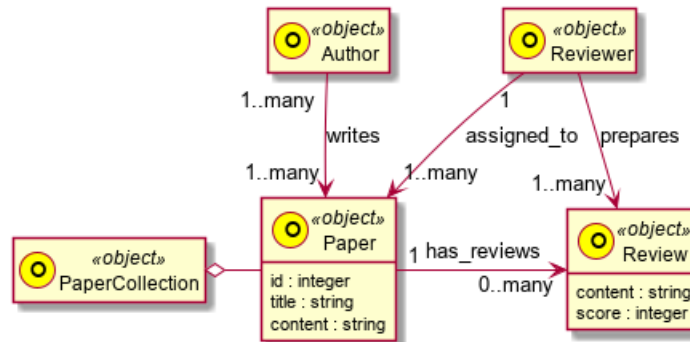
CMS ma być systemem wieloagentowym, który wspomaga proces zbierania, recenzowania i wydawania publikacji przesyłanych na konferencje (naukowe). Sam proces wymaga współpracy i koordynacji wielu osób i rozpoczyna się zgłaszaniem prac przez autorów oraz rejestracją recenzentów. Po upłygnięciu terminu zgłoszeń komitet programowy (*program committee, PC*) przydziela poszczególne prace recenzentom w celu przygotowania recenzji. Po przygotowaniu recenzji PC podejmuje decyzje o przyjęciu lub odrzuceniu poszczególnych prac. Autorzy są informowani o decyzji i – w przypadku akceptacji – proszeni o przygotowanie ostatecznej (poprawionej) wersji pracy. Poprawione wersje są zbierane, a następnie wysyłane do drukarni w celu przygotowania materiałów konferencyjnych (*proceedings*).

# Proces O-MaSE

- Złożona wersja procesu
- Stworzenie następujących modeli
  1. Model dziedziny
  2. Model celów (podstawowy i rozszerzony)
  3. Model organizacji
  4. Model ról
  5. Model klas agentów
  6. Modele protokołów
  7. Modele planów agentów
  8. Model polityki

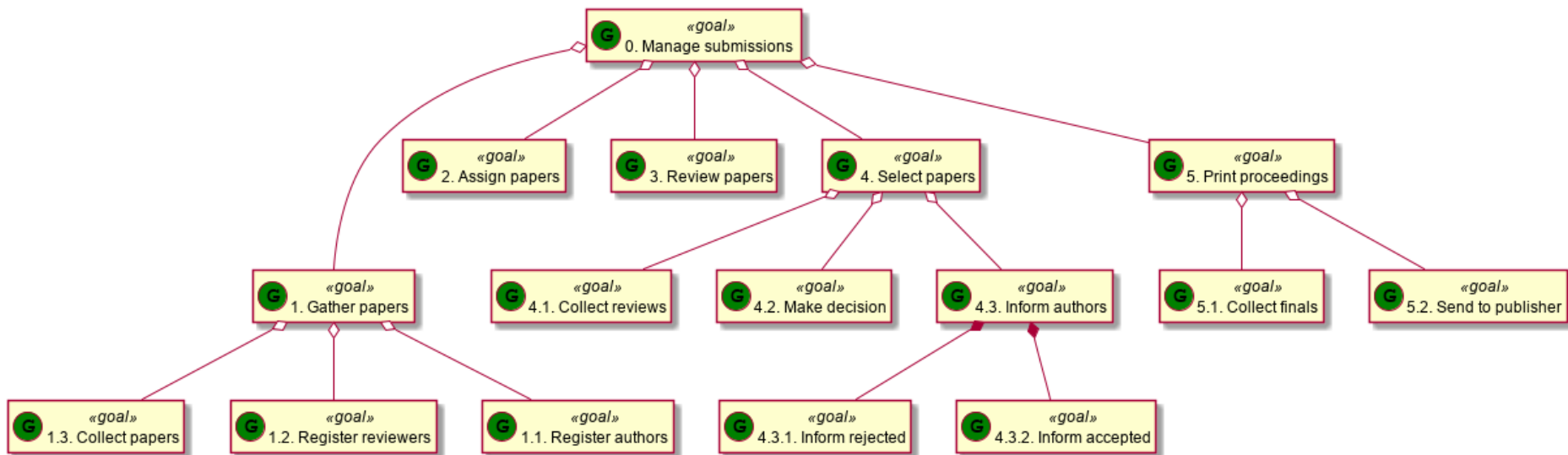
# Model dziedziny

- Obiektowy model danych przetwarzanych przez system
- Klasy zdefiniowane w modelu pojawiają się jako parametry celów, komunikatów, obliczeń...



# Podstawowy model celów

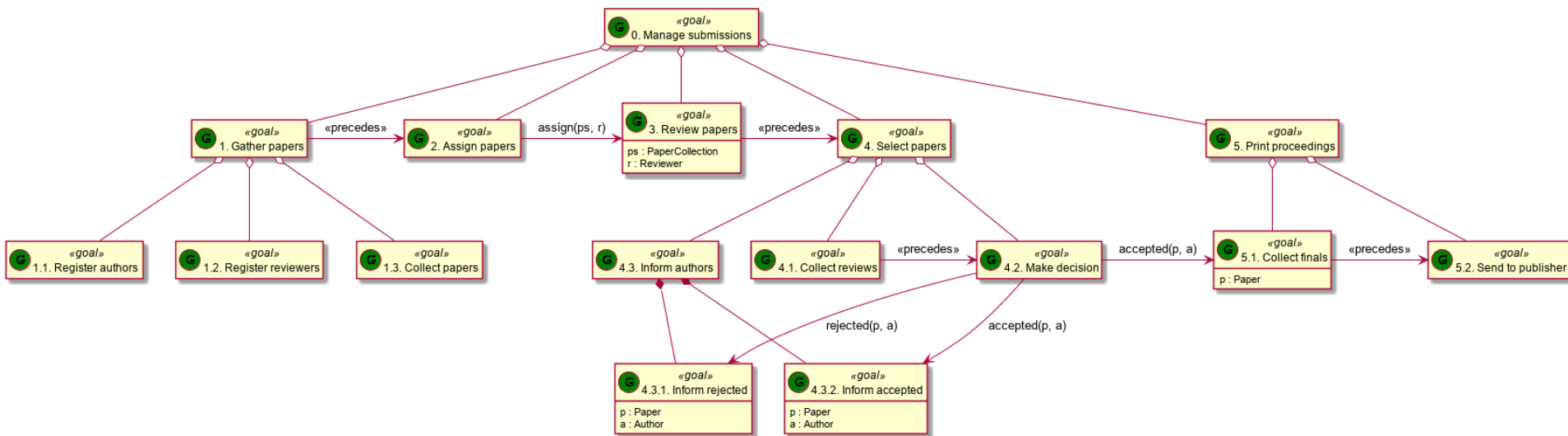
- Dekompozycja celu głównego
- AND/OR w celu określenia warunków spełnienia celu





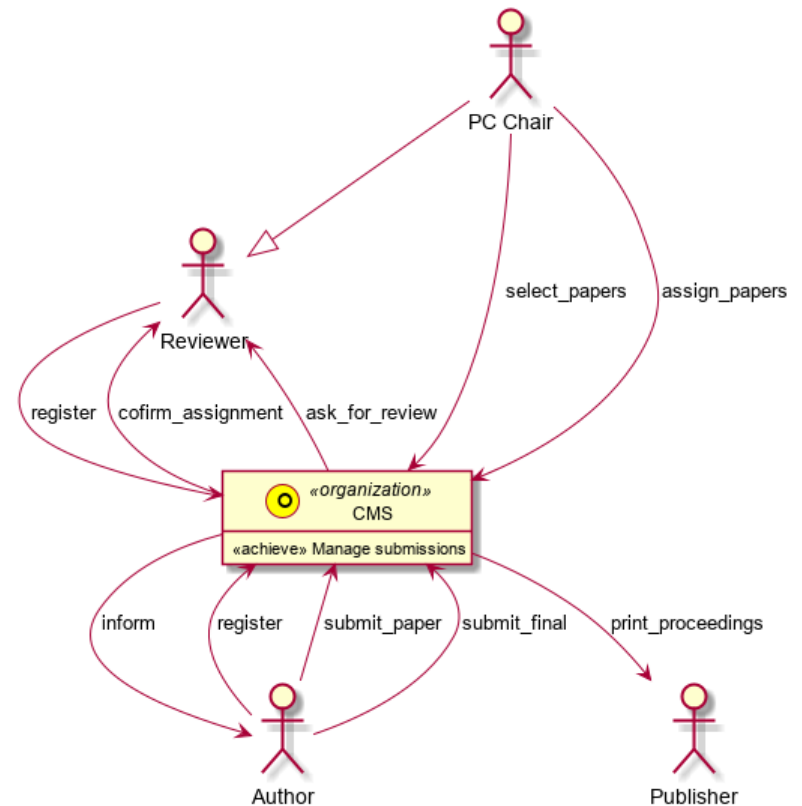
# Rozszerzony model celów

- Uwzględnienie dynamiki systemu (zależności między celami) oraz parametryzowanych celów



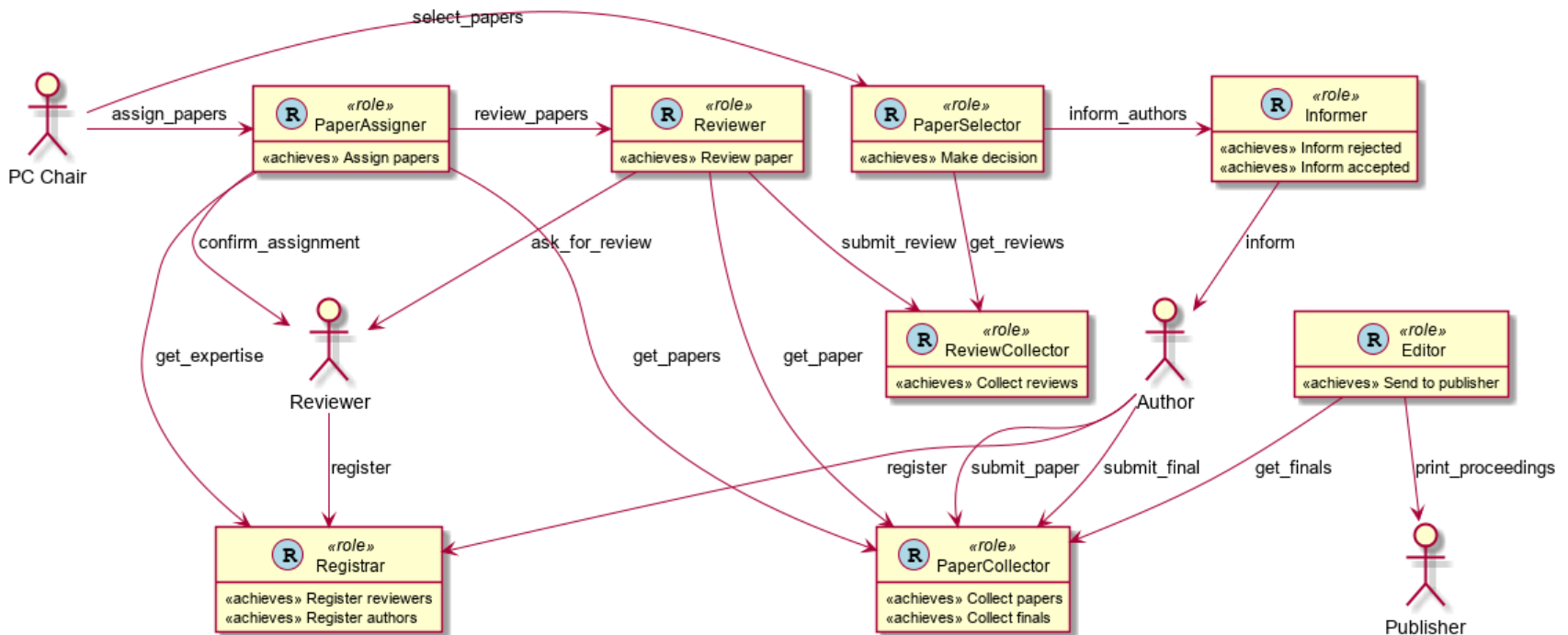
# Model organizacji

- Identyfikacja „interfejsów” systemu z zewnętrznymi aktorami (użytkownikami lub innymi systemami)
- Interakcje opisane za pomocą protokołów
- Możliwość wskazania zależności (relacja *is-a*) między aktorami



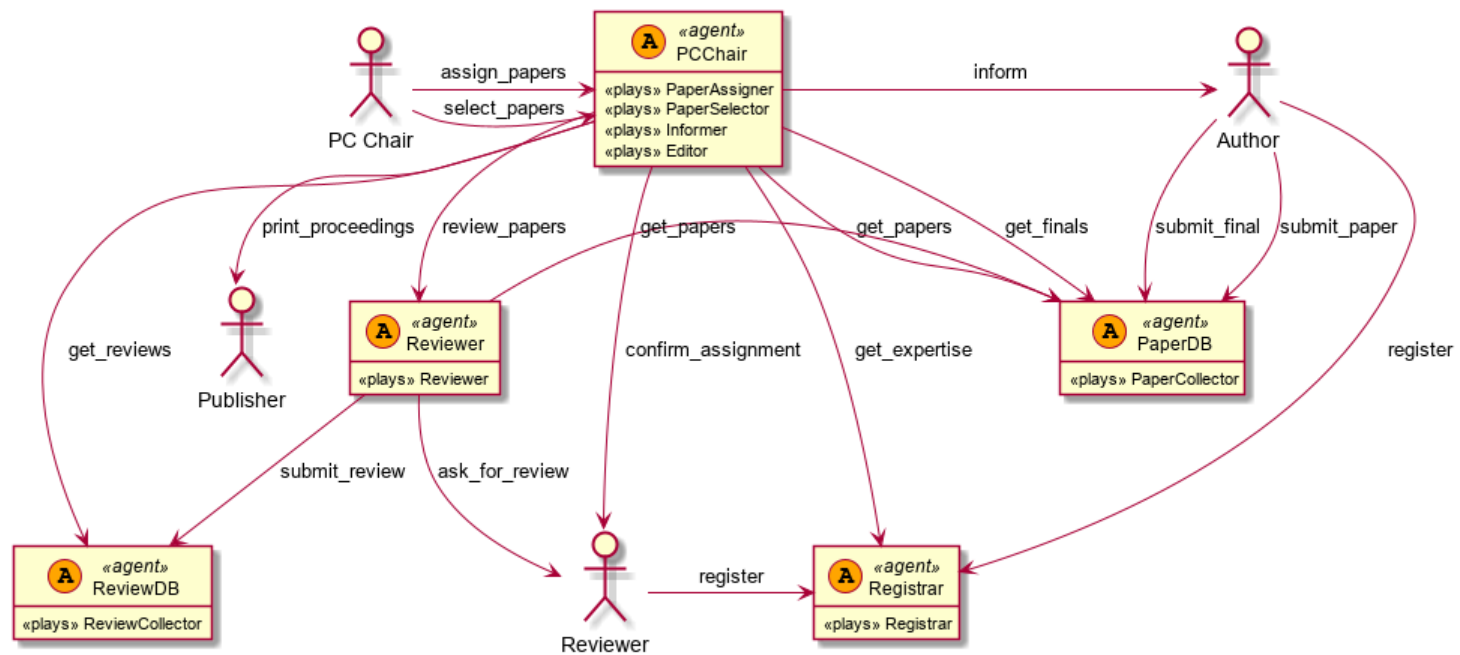
# Model ról

- Przepisanie „atomowych” celów z modelu celów (liści) do poszczególnych ról
- Uwzględnienie wprowadzonych aktorów i protokołów



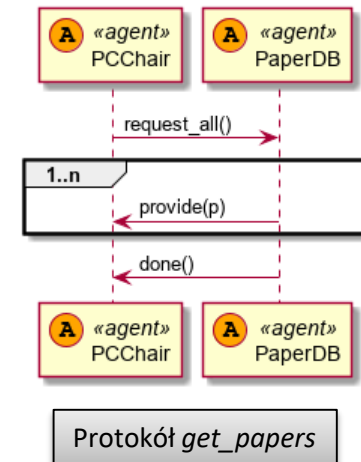
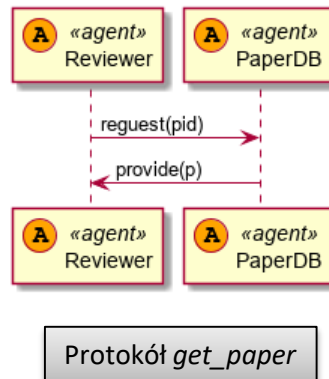
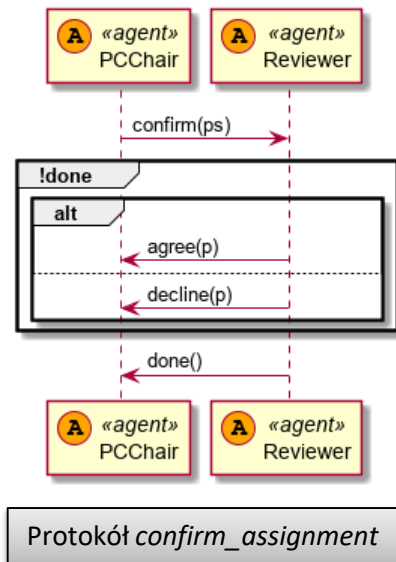
# Model klas agentów

- Przepisanie ról do (klas) agentów
- Agregacja i usuwanie protokołów przy łączeniu wielu ról w ramach jednej klasy agenta



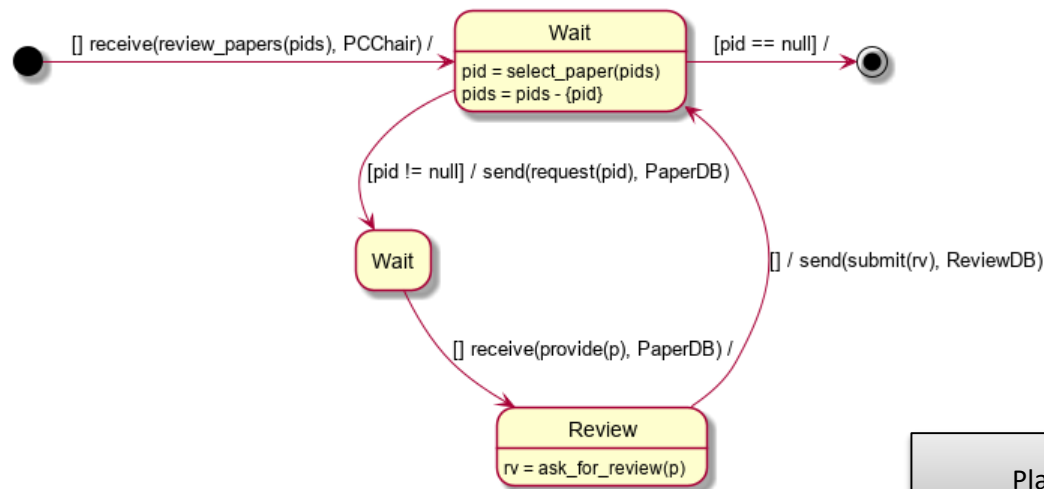
# Modele protokołów

- Modele dla wszystkich protokołów z modelu klas agentów



# Modele planów

- Modele planów dla wszystkich klas agentów (ról lub celów)
- Warunkowe przejścia między stanami, sekwencyjne wykonywanie akcji w poszczególnych stanach



Plan dla agenta *Reviewer*

# Model polityki

- Reguły kontrolujące wartości obiektów z modelu dziedzinowego oraz zachowanie się agentów
- Wyrażone w logice pierwszego rzędu – możliwość stosowania mechanizmów wnioskowania

$$\forall p1, p2: Paper \ p1.ID = p2.ID \Rightarrow p1 = p2$$

$$\forall p: Paper, r: Reviewer, a: Author \ a = r \wedge writes(a, p) \Rightarrow \neg assignedTo(p, r)$$

$$\forall p: Paper, r: Reviewer \ submitReview(r, p) \Rightarrow assignedTo(p, r)$$

# Zadanie na dziś



Proszę zastosować metodę O-MaSE do przygotowania formalnego projektu (zestawu modeli) systemu do handlowania książkami z poprzednich zajęć. Projekt powinien zawierać model dziedziny\*, celów, klas agentów, protokołów i planów.

Możemy założyć, że agent w momencie uruchomienia ma przydzielony zestaw książek, a agent sprzedający jest podczas działania systemu kontrolowany przez użytkownika.