

# Мультиагентно-орієнтоване програмування

3-й рівень навчання, осінь 2021

- Доц. Баклан І.В.
- Email: [iaa@ukr.net](mailto:iaa@ukr.net)
- Web: [baklaniv.at.ua](http://baklaniv.at.ua)

# Лекція 1

## Введення до мультиагентно-орієнтованого програмування

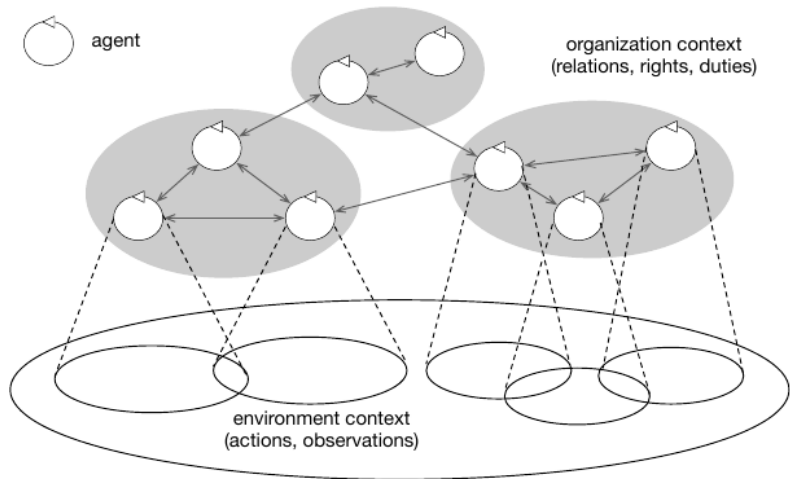
Багатоагентні системи - це *парадигма* моделювання та інженерії складних систем. Під парадигмою ми розуміємо набір концепцій, прийомів, технологій та методологій. Моделювання означає створення формалізованих уявлень (моделей) на основі цих концепцій, які охоплюють істотні аспекти, що стосуються структури та поведінки цільових систем. Моделювання є важливим аспектом двох основних контекстів, у яких використовуються MAS: (1) моделювання, в якому моделі корисні для опису та моделювання існуючих складних систем, природних чи штучних, для аналізу їх властивостей; та (2) інженерія, яка більше орієнтована на проектування та розробку систем та додатків.

Прикладом для імітаційного випадку є агентна модель для моделювання дорожнього руху, в якій автомобілі моделюються як агенти, а міські та вуличні знаки є частиною середовища агентів. Прикладом інженерного випадку в тій же області є модель розумного міста, в якій автономні безпілотні автомобілі розроблені як агенти, що взаємодіють з іншими агентами та цифровими службами, що представляють інтелектуальну інфраструктуру. Інженерна перспектива - це та, що прийнята в цих лекціях.

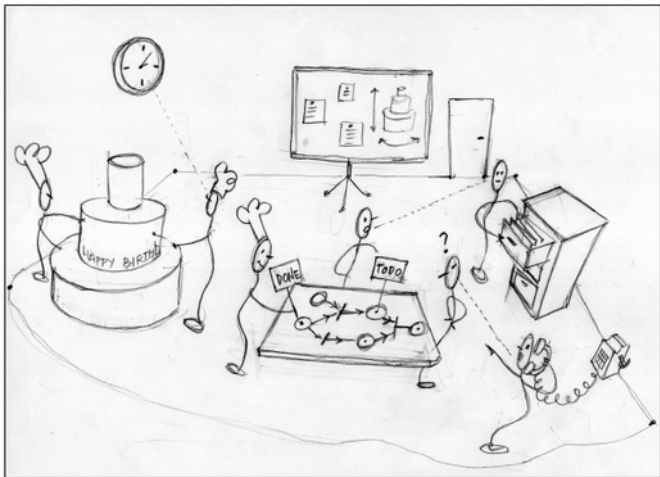
Використовуючи MAS, складна система моделюється (і проектується, і програмується) як організація автономних агентів, розташованих і взаємодіючих у деякому (логічному) середовищі. Агенти представляють суб'єкти прийняття рішень систем; тобто агенти - це суб'єкти, призначені для автономного досягнення певної мети, інкапсулюючи для цієї мети логічний потік управління та приймаючи рішення про те, як поводитися та взаємодіяти. У прикладі розумного міста метою агента, що представляє автономний безпілотний автомобіль, може бути досягнення певного місця призначення, вибір найбільш зручного шляху відповідно до уподобань користувачів. Агенти логічно розташовані в середовищі, яке вони сприймають і діють для досягнення своїх цілей. Середовище являє собою контекст, в якому діє агент.

У прикладі розумного міста середовищем може бути як фізичне середовище, таке як вулиця, так і цифрові послуги, такі як спільні розміщені дошки оголошень, що передають відповідну інформацію.

Середовище може бути масштабним і розповсюдженим; тим не менш, агент може спостерігати та діяти лише на його частину в будь-який момент часу (див. рисунок 1.1). Аналогія з реальним світом досить проста: люди (тобто агенти) сприймають і діють у певному середовищі, вирішують, що робити для досягнення своїх цілей, взаємодіють з іншими людьми тощо. Як і у випадку людини, навколишнє середовище можна розглядати також як набір ресурсів та інструментів, якими агенти можуть ділитися та використовувати їх для виконання своїх завдань. На малюнку 1.2 наведено приклад пекарні, яку ми також використовуємо в наступних лекціях.



Мал.1.1 Представлення багатоагентних систем, за Дженнінгсом (2001).



Мал.1.2 Аналогія між MAS та людською системою (сценарій пекарні).



Багатоагентна система включає в себе кілька агентів, які спілкуються за допомогою деякої мови спілкування агентів високого рівня (ACL) і співпрацюють для досягнення спільних цілей. Організація (малюнок 1.1) чітко фіксує основні аспекти, що характеризують завдання (та функціональні можливості, поведінку та властивості) системи в цілому. У прикладі розумного міста, наприклад, набір агентів, що представляють автономні безпілотні автобуси, може утворити групу як з окремими місіями, так і спільними цілями, а також з правами та обов'язками, які обмежують їхню поведінку та взаємодію всередині організації та між організаціями.

## Парадигма MAS проти існуючих парадигм

Агентно-орієнтоване моделювання поділяє деякі характеристики з об'єктно-орієнтованим моделюванням. Обидва дотримуються принципу приховування інформації та визнають важливість взаємодії. Як і об'єкти, агенти приховують внутрішній стан та деталі реалізації, а передача повідомлень на основі певної мови спілкування спільних агентів є основним засобом для спілкування агентів між собою.

Перша ключова відмінність полягає в тому, що агенти інкапсулюють логічний потік управління, який робить їх активними (а не пасивними подібними об'єктами), вбудовуючи логічний потік управління. На відміну від об'єктів, агенти інкапсулюють стан, поведінку та контроль цієї поведінки. Це також впливає на модель спілкування/взаємодії, яка є строго асинхронною, так що агент, який надсилає повідомлення іншому агенту, не передає свій потік управління (як у випадку виклику методу між пасивними об'єктами), а елемент управління залишається інкапсульованим.

Аналогічно агент обробляє повідомлення зі своїм власним потоком управління.

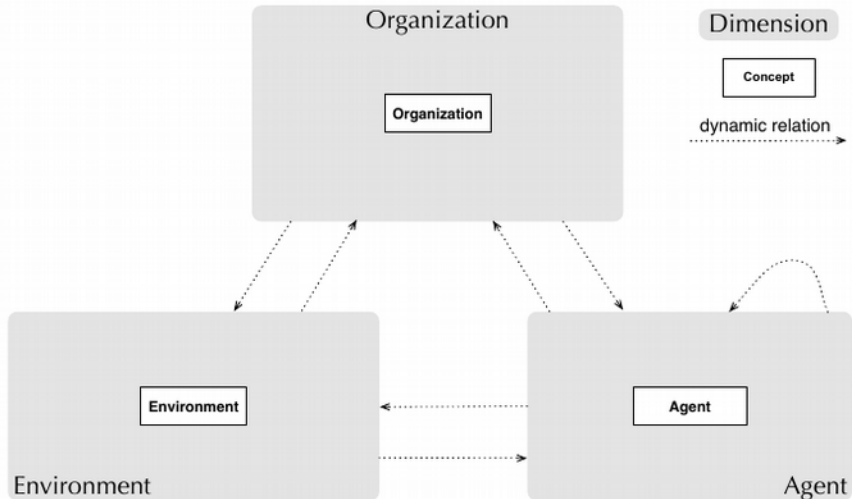
З цієї точки зору агенти схожі на акторів. Однак актори є суворо реагуючими суб'єктами; вони діють лише у відповідь на отримання повідомлення. Після виконання обробника (або методу), пов'язаного з повідомленням, якщо немає інших повідомлень для обробки, актор стає неактивним. І навпаки, агенти є цілеспрямованими проактивними утвореннями, тому вони діють для того, щоб виконувати свої завдання, навіть якщо повідомлення не надходять.

Ще однією головною відмінністю між агентно-орієнтованим моделюванням та об'єктно-акторським моделюванням є середовище як першокласна абстракція. У чисто об'єктно-орієнтованому світі програмування все моделюється як об'єкт. У чистому акторському світі все моделюється як актор. В агентно-орієнтованому моделюванні агенти спілкуються з іншими агентами, але також взаємодіють із оточенням за допомогою дій та сприйняття. Отже, агентно-орієнтоване моделювання не передбачає моделювання всього як агента. Існує поділ між проблемами між тими частинами системи, що представляють/інкапсулюють прийняття рішень, та тими частинами системи, що представляють суб'єкти, які підлягають контролю, забезпечуючи дії та спостережуваний стан/події для цієї мети.

## Мультиагентно-орієнтоване програмування

В принципі, будь-яка технологія програмування може бути використана для реалізації MAS, якщо дати чіткий опис моделі. Однак ризик полягає в агентурній інтерпретації малюнка 1.1, у якій середовище та/або контекст організації представлені та управляються у свідомості агентів. Прийняття мов програмування, які безпосередньо забезпечують першокласні абстракції програмування, значно спрощує наше завдання, даючи можливість підтримувати рівень абстрактності узгодженим від часу розробки до часу розробки, а також під час виконання.

Багатоагентне орієнтоване програмування-це підхід до програмування MAS, який сприяє використанню першокласних абстракцій програмування, що стосуються трьох основних вимірів, що характеризують багатоагентну систему, а саме: виміру агента, виміру середовища та організаційного виміру (показано на малюнку 1.3). Кожен вимір визначає набір концепцій та першокласних абстракцій, які відображають різні проблеми MAS.



Мал.1.3 Виміри багатоагентно-орієнтованого програмування



*Вимір агента* групує концепції та абстракції програмування для визначення та програмування агентів, які беруть участь у системі. Концепція агента - це ключова абстракція, яка використовується для програмування суб'єктів, що приймають рішення, здатних забезпечити локальну гнучкість та динамічність, реагуючи на події, одночасно проактивно спрямовуючи свою поведінку на досягнення майбутніх станів системи з метою досягнення певних *цілей*. У результаті програмне забезпечення має власну логічну нитку контролю для автономного досягнення цих цілей, взаємодіючи з середовищем, іншими агентами та організацією, яка регулює загальну систему.

Однією з найвизначніших особливостей агентів у цьому погляді є *автономія*. Автономія невіддільна від концепції агента в тому сенсі, *що* агент покликаний міркувати про те, чого слід досягти, і, *що* важливо, *як* це зробити з огляду на поточні обставини системи. Для цього агент має проактивність, тобто здатність проявляти ініціативу щодо дій, які необхідно виконати для досягнення своїх цілей; *реактивність*, тобто здатність оперативно адаптувати свою поведінку залежно від подій, сприйнятих з навколишнього середовища; і *соціальні здібності*, тобто здатність спілкуватися та співпрацювати з іншими агентами. На рівні організації автономія передбачає децентралізацію контролю, так що на агента (та його поведінку) зазвичай впливають, але не регулюються ні іншими агентами, ні організаційними нормами. Норми використовуються для визначення очікуваної поведінки в соціальній системі, але автономні агенти можуть вирішити не дотримуватись.

*Вимір середовища* пропонує концепції та першокласні абстракції для визначення та програмування розподілених ресурсів та зв'язків із реальним світом, що ділиться між агентами. Якщо агенти корисні для моделювання автономних цільових орієнтацій, середовище як першокласна абстракція корисно для моделювання будь-яких елементів, які можуть бути використані або контрольовані агентами для досягнення своїх цілей. Абстракція середовища - це те, що робить агентів розташованими, тобто логічно поміченими в контекст, який надає їм набір дій, що впливають на навколишнє середовище, і викриває якийсь спостережуваний стан та події, які агенти можуть сприймати.

*Вимір організації* збирає всі необхідні концепції, беручи участь у визначенні та програмуванні відносин, спільних завдань та політики серед агентів, що взаємодіють у спільному середовищі. Центральне поняття організації визначає структурування, координацію та регулювання спільної роботи агентів. Віддзеркалюючи функцію автономності виміру агента, що стикається з динамічністю виміру середовища, найбільш визначальними з цієї точки зору є особливості координації та регулювання, які допомагають задовольнити вимогу відкритості. Координація відноситься до підтримки роботи багатьох агентів, які залежать один від одного для досягнення того, чого вони прагнуть досягти окремо або колективно. Регламент відноситься до підтримки приборкання автономії агентів-учасників.

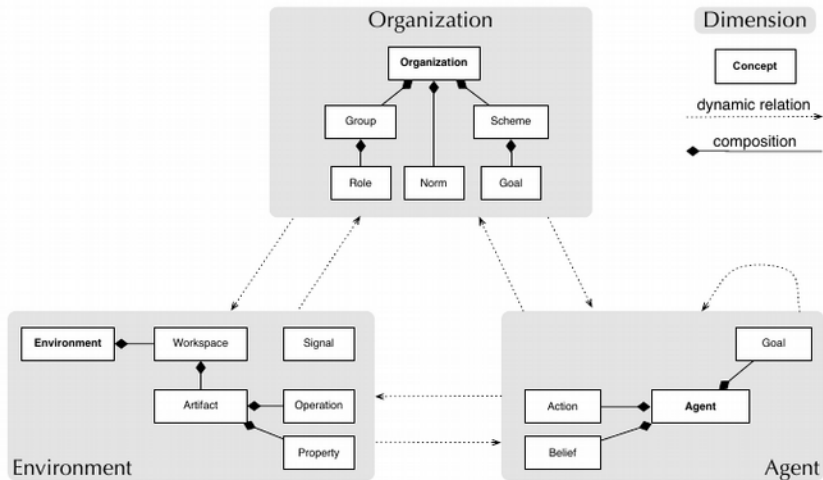
Регулятивний аспект, як правило, здійснюється з використанням норм (соціальних правил, яких очікується, що агенти будуть дотримуватись і можуть бути якимось чином покарані за їх невиконання).

Програмування організації відкриває для агентів можливість міркувати про свої відносини, спільні завдання та політику, а також вирішувати питання про їх адаптацію, а також про відповідність обмеженням, що впливають з цього.

## Деякі визначення

Кожен вимір МАОП включає різноманітні концепції та абстракції програмування, які підтримують розвиток MAS. Тут ми надаємо вид з висоти пташиного польоту, призначений для карти (див. малюнок 1.4), спочатку визначаючи основні абстракції для кожного виміру, а потім зазначаючи зв'язки, що склеюють розміри.

У наступних лекціях ми більш детально розглянемо розміри.



Мал.1.4 Виміри і основні поняття МАОП

**Вимір агента.** У вимірі агента концепція *мети* дуже важлива; він забезпечує засоби для представлення певного стану справ, до якого агент хоче донестись. Чітке уявлення про довгострокові цілі, яких мають досягти агенти, має важливе значення для автономності та активності. У деяких випадках це відповідає поясненню цілей проектування кожного агента. Так само важливим для автономії є те, що агенти здатні приймати раціональний вибір щодо того, які цілі переслідувати під час виконання, а також вибирати засоби, які використовувати для досягнення цих цілей.



Цей вибір та подальша поведінка керуються *переконаннями* агентів. Переконання - це просто явні уявлення про доступну агенту інформацію, на якій агент може міркувати; цей термін використовується для того, щоб підкреслити, що інформація, доступна агенту, може бути неправильною та/або неповною у типових багатоагентних системах. Переконання набуваються шляхом сприйняття стану навколишнього середовища та спілкування з іншими агентами, наприклад. Основним моментом агента є дії, які він вживатиме для зміни стану навколишнього середовища для досягнення своїх цілей, або для взаємодії з іншими агентами чи елементами середовища.

**Вимір середовища** У вимірі середовища поняття робочого простору використовується для визначення топологічних або символічних областей середовища, які заповнені набором артефактів та агентами. Артефакт являє собою реальний або концептуальний ресурс середовища за допомогою набору операцій, які агенти можуть використовувати для виконання дій, та властивостей, які агенти можуть спостерігати для набуття переконань. Нагадаємо, що суб'єкти навколишнього середовища не є автономними чи ініціативними, як агенти. Використовуючи ці операції над екземплярами артефактів, наявними в даний час у системі, агенти можуть змінювати стан середовища, а також спостерігати за ним, враховуючи властивості, які надають ці екземпляри артефактів. Артефакти корисні для модуляції середовища, роблячи його динамічним - артефакти можуть створюватися і знищуватися динамічно (агентами).

За допомогою артефактів середовище забезпечує рівень абстракцій програмного забезпечення для підтримки взаємодії агента за допомогою спільних ресурсів неявним чином контрольованим способом.

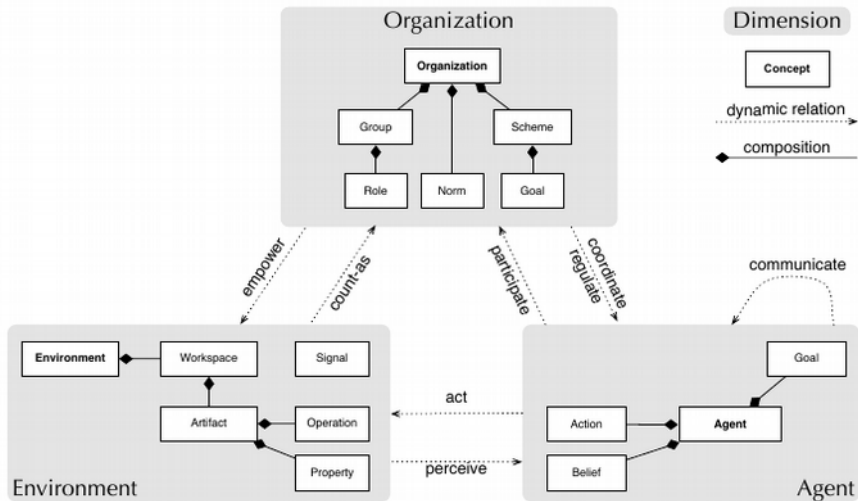
**Організаційний вимір** В організаційному вимірі концепція групи використовується для забезпечення соціальної структури системи, надаючи підтримку у визначенні очікуваної скоординованої поведінки в системі, а також прав та обов'язків, які повинні виконувати агенти. У групі ролі визначають взаємодію та відносини, що відбуваються в групі. Вони також беруть участь у визначенні прав та обов'язків, виражених як норми. Очікувана поведінка щодо прав та обов'язків виражається як набори організаційних цілей, що використовуються у визначенні соціальних планів, які є деревами розкладання цілей, що містяться у соціальних схемах, які виконуються під відповідальність груп. Соціальний план чітко визначає очікуване скоординоване досягнення цілей при виконанні прав та обов'язків, виражених нормами.

Коли організації нав'язуються суспільству агентів, агенти, які відіграють роль у призначених групах, повинні працювати разом, координуючи свої дії для досягнення організаційної мети системи, дотримуючись прав та обов'язків, покладених нормами, що відповідають до своїх ролей.

Як ключовий аспект підходу MAOP, усі ці абстракції зберігаються під час виконання платформою, на якій працює MAS (у нашому випадку JaCaMo). Таким чином, програмна система, запрограмована за допомогою MAOP, може динамічно змінюватись і реорганізовуватись різними способами, наприклад, за допомогою створення під час виконання організації та артефактних екземплярів у середовищі, а точніше агентів, що приєднуються до організацій та виходять із них через ті ролі, які вони вирішують прийняти. Це корисно для багатьох функцій, які потрібні нам у сучасних сучасних системах, про що ми обговоримо згодом у цій лекції.

## Узагальнений погляд

Таким чином, об'єднання вимірів для програмування системи призводить до створення багатоагентної системи, яка являє собою набір агентів, одне середовище та набір взаємодіючих організацій, як показано на малюнку 1.5. Взаємодія є важливим аспектом багатоагентного підходу до програмування. Багатоагентні системи-це взаємодія між автономними агентами, які спілкуються між собою, та взаємодія між ними та навколишнім середовищем, за допомогою яких вони також взаємодіють з організаціями, які регулюють та координують частину їх діяльності. Ці взаємодії відбуваються під час виконання системи і тому представлені на малюнку 1.5 у вигляді динамічних співвідношень між вимірами.



Мал.1.5 Динамічні зв'язки між екземплярами концепцій вимірів МАОП.



Динамічні зв'язки (малюнок 1.5) пов'язують виміри агента, середовища та організації, породжуючи замкнутий і багатий цикл взаємодіючих екземплярів концепцій, що належать до різних вимірів.

Агенти можуть вирішувати спілкуватися з іншими агентами, створюючи між ними динамічні стосунки, роблячи їх здатними взаємодіяти з іншими агентами, особливо з іншими агентами, які наразі беруть участь у тих самих організаціях, але не виключно. Те, як відбувається ця пряма взаємодія агента з агентом, ґрунтується на теорії мовленнєвих актів, тому комунікація розглядається як дія, що змінює психічний стан (наприклад, переконання та цілі) як відправника, так і одержувача. Таким чином, ця взаємодія є досить особливою у порівнянні з іншими підходами до спілкування в розподілених обчислень.

Детальніше це обговорюється у лекції 4.

Взаємодія у багатоагентній системі не обмежується лише спілкуванням між агентами; це також відбувається між агентами та оточенням. Як згадувалося раніше, агенти можуть сприймати (тобто спостерігати або відчувати) артефакти, що знаходяться в навколишньому середовищі, і реагувати на це сприйняття. Агенти також можуть діяти, змінюючи артефакти їхній стан. У робочій області агенти та артефакти є видимими один для одного, хоча агенти можуть ігнорувати або зосереджуватися на певних артефактах за власним бажанням, щоб допомогти у масштабуванні.

Агенти взаємодіють з навколишнім середовищем, а також з іншими агентами. За допомогою середовища агенти можуть побічно взаємодіяти між собою. Наприклад, агент виконує дію відкриття дверей у своєрідному артефакті дверей, який спостерігається іншим агентом, який сприймає та представляє його як віру в навколишнє середовище і може викликати ціль, яка веде агента до кімнати.

Зміни стану навколишнього середовища також можуть зараховуватися як зміни стану організації. Щоб підтримувати спільну роботу агентів, організація повинна бути уважною до того, що агенти роблять у навколишньому середовищі. Наприклад, якщо існує залежність між завданнями двох агентів, коли перше завдання сприймається як досягнення певної мети через стан навколишнього середовища, організація може вимагати від агента, відповідального за друге завдання, здійснити певний курс дій, щоб виконати друге завдання, як воно раніше зобов'язалося. Це один із способів, яким організація може регулювати та координувати діяльність агентів.

І навпаки, організації, що створюють організаційний вимір, можуть розширити можливості середовища, дозволивши їм контролювати та регулювати дії чи сприйняття агентів. Це динамічне відношення є практичним способом розташування організацій у середовищі, як це відбувається з агентами, регулюючи певну частину середовища (наприклад, світлофор на перехресті) певним чином та керуючи ним по-різному в інших частинах.

Нарешті, динамічні відносини між вимірами агента та організації стосуються впливу, який організації можуть мати на агентів: вони можуть регулювати та координувати агентів для динамічно створюваних завдань. Координатне відношення відноситься до управління залежностями між діями, що здійснюються агентами. Відношення регулювання, у свою чергу, відноситься до здійснення контролю за цією діяльністю. Звичайно, оскільки агенти є автономними, це може статися лише в тому випадку, якщо вони активно вирішать брати участь в одній або кількох існуючих на даний момент організаціях у системі.

## Подолання проблем

**Автономія** MAOP пропонує структурований підхід до автономії з чітким розділенням проблем. Перш за все, агенти - це місце, де розглядається автономія, тоді як артефакти, які є активними або пасивними сутностями, що знаходяться в оточенні, вважаються неавтономними. Концепції виміру агента підтримують автономію, дозволяючи дизайнерам зосереджуватися на чітко представлених цілях, а агентам - міркувати про найбільш підходящі варіанти дій для їх досягнення. Крім того, різні методи штучного інтелекту можуть бути безпосередньо підключені до архітектури агента, щоб збільшити автономність.

Розділення проблем між агентами, автономних утворень та артефактів, неавтономних утворень, приймає все своє значення у визначенні організації, яка використовує абстракції для моделей координації та регулювання, спрямованих на автономні утворення. Це дозволяє декільком автономним агентам координувати свої дії, тим самим у певному сенсі приборкуючи їхню автономність, щоб система працювала злагоджено. Таким чином, MAOP підтримує програмування автономних систем з виміром агента, що стосується визначення самих автономних утворень, виміром середовища, націленим на визначення спільних сутностей, які є джерелами сприйняття та цілями дій автономних утворень, і, нарешті, з організаційним виміром виражаючи необхідний контроль автономних утворень у спільному середовищі.



**Децентралізація та розподіл** Взаємодія є центральним механізмом у МАОП. Це сприяє слабкій взаємозв'язку між автономними утвореннями і таким чином забезпечує можливість розробки та розгортання децентралізованих систем. З одного боку, використання складних прямих взаємодій між агентами за допомогою мовленнєвих актів дозволяє автономним агентам діяти на інших автономних агентів. З іншого боку, використання непрямих взаємодій дозволяє незалежності агентів, які використовують середовище як спільне середовище взаємодії між ними. За допомогою організацій, які дозволяють декларувати програмування шаблонів координації та регулювання, можна контролювати та структурувати такі децентралізовані системи. Навіть якщо ці шаблони поділяються між агентами, центральна контрольна точка не вводиться.

Процес прийняття рішень може бути децентралізованим. Організації вирішують нові виклики у розробці програмного забезпечення: вміти справлятися з координацією роботи великої кількості автономних агентів (як програмного забезпечення, так і людей) та підтримувати децентралізацію такої координації.

Доповнюючи децентралізацію, підхід MAOP надає багато можливостей для розповсюдження. Завдяки модульності, що сприяє розділенню проблем, модульними компонентами, які можна розглядати з точки зору MAOP, є, крім агентів, робочих просторів та артефактів у вимірі навколишнього середовища та організацій, груп та схем у організаційному вимірі. Таким чином, розповсюдження може стосуватися запуску на різних машинах агентів, робочих областей, артефактів, організацій, груп, схем та інших компонентів. Кілька організацій з кількома запущеними агентами та різними частинами середовища можуть бути розподілені на різних обчислювальних платформах, що також допомагає підтримувати масштабованість, можливо, "святий Грааль" сучасних обчислень.

**Відкритість** Вирішення проблеми відкритості за допомогою підходу MAOP допомагає визначити та структурувати відповідні відповіді на різні еволюції агента (наприклад, вхід/вихід агентів), середовище (наприклад, створення/видалення артефактів та топологію робочих просторів) та організації (наприклад, зміни у шаблонах координації або регулювання в організації та створення/видалення організацій) виміри. У MAOP агенти, артефакти, робочі простори та організації з їх групами та схемами призначені для створення, виявлення та, можливо, утилізації самими агентами під час виконання. Це базовий спосіб, яким MAOP підтримує динамічну розширюваність (крім модульності) агентів, середовища та організацій. Наприклад, новий агент може бути здатним вписатися у глобальне функціонування, що є результатом різних взаємодій системи з точки зору агентів, артефактів та організацій.

Агенти можуть впоратися з неповними знаннями та контролем і можуть взаємодіяти з іншими агентами, з артефактами навколишнього середовища або з нормами організації, які не були відомі під час розробки чи розгортання. Система може адаптувати своє глобальне функціонування до несправності або припинення роботи агента. Він може бути готовий мати справу з будь-якою кількістю агентів. Навколишнє середовище та його артефакти можуть еволюціонувати та інтегрувати поступово нові функціональні можливості (наприклад, ресурси зберігання та обробки) шляхом розгортання нових артефактів. Організації можуть еволюціонувати та інтегрувати нові норми, нові соціальні плани та нові структури відповідно до дій агентів.

Зіткнувшись з безперервною еволюцією, що виникає від сутностей, які можуть бути невідомими під час ініціалізації системи, підхід MAOP пропонує різні засоби для підтримки та забезпечення цілісної поведінки системи. Явні та декларативні уявлення організацій допомагають регулювати та контролювати такі відкриті системи, визначаючи межі поведінки як допустиму поведінку, що відповідає стратегіям координації та регулювання, що використовуються в системі. Крім цього м'якого контролю, абстракції навколишнього середовища обмежують репертуар дій, які можна здійснювати у фізичному середовищі за допомогою артефактів. Вони допомагають визначити межі поведінки як можливу поведінку, відповідну діям, запропонованим спільним сховищем та обробкою артефактів, а також механізмам моніторингу та виявлення сутностей системи.

Таким чином, можна гарантувати, що поведінка, вироблена агентами, навіть якщо вона розвинена іншими зацікавленими сторонами, ніж зацікавленими сторонами багатоагентної системи, належить до набору можливої та допустимої поведінки. З іншої точки зору, він також підтримує зміну та адаптацію стратегій координації та регулювання чи блоків зберігання та обробки без зміни агентів.

**Неоднорідність** Кожен із орієнтованих на багато агентів вимірів моделювання запроваджує засоби для визначення різних модулів, які вводять неоднорідні уявлення та динаміку на системному рівні, які потенційно могли бути запрограмовані різними людьми або які дійсно представляють інтереси різних, можливо конкуруючих компаній.

Загалом, агенти в системі мають різні можливості, можливо, різні архітектури та інші відмінності. Артефакти інкапсулюють різні ресурси, які можуть бути фізичними або цифровими. Організації виражають різні моделі координації та регулювання. Тому, як правило, MAS за підходом MAOP дуже неоднорідний.



Як уже згадувалося раніше, щоб зіткнутися з неоднорідністю, сумісність є затребуваною властивістю. Відповідно до IEEE, «сумісність означає здатність двох або більше систем чи компонентів обмінюватися інформацією та використовувати інформацію, якою обмінювались» (Geraci et al. 1991). Було запропоновано кілька варіантів загальної ідеї сумісності (наприклад, Morris et al. 2004; Tolk and Muguira 2003), щоб поширити її на ширший комплекс проблем: враховуючи технічні та бізнес-сумісність, будь -яку взаємодію між суб'єктами господарювання (будь то люди , програмне забезпечення, речі або змішане населення) вимагає спільного або спільного уявлення про навколишнє середовище та концепцій, що використовуються в *організації* та *регулюванні* цього середовища.

Неоднорідність часто пов'язана з побоюваннями щодо сумісності різних систем. У МАОП проблеми сумісності визначаються та розподіляються відповідно до інтерфейсів між кожним із вимірів МАОП.

До них належать такі приклади:

- Agent-Agent, який визначає загальні мови спілкування для прямої взаємодії між гетерогенними агентами;
- Агент-Організація, що визначає представлення організації, щоб дозволити агентам читати, діяти та міркувати про моделі координації та регулювання, що регулюють їх поведінку; та
- Середовище агента, яке визначає чіткі посібники з використанням репертуару дій та сприйняття, наявних у артефактах середовища, щоб дозволити агентам читати, використовувати та міркувати про доступні дії.

**Адаптованість** Для вирішення проблеми адаптованості підхід МАОП дозволяє розробнику ідентифікувати та вирішувати проблему в різні часові та тривалі терміни, що мають місце у багатоагентній системі: короткочасна повторювана діяльність, що складається з дії та сприйняття, що пов'язують агентів із їхнім оточенням, довгострокова діяльність, що відповідає управлінню цілями в агентах щодо прийняття рішення про їх дії, та середньострокова діяльність, представлена в організації, що визначає стратегії та політику, які контролюють та регулюють управління цілями та рішеннями у агентів, а отже, їх дії у спільне середовище.

Прийнята модель програмування агентів, яка базується на архітектурі *переконань-бажань-намірів (Belief-Desire-Intention - BDI)*, дозволяє вибирати та приймати різні плани для досягнення однієї і тієї ж мети залежно від контексту, що дозволяє легко розробити контекстно-усвідомлену поведінку. Виконання плану може бути перерване динамічно, наприклад, у разі невдач або змін у середовищі, що вимагають прийняття різних стратегій або навіть різних цілей.

Використання першокласних абстракцій для представлення, модуляції та маніпулювання навколишнім середовищем дає змогу агентам самостійно приймати рішення щодо адаптації набору ресурсів та інструментів для використання для досягнення своїх цілей.

Поєднання середовищ та організаційних вимірів забезпечує усвідомлення контексту та адаптованість на рівні організації. Що стосується агентів, то стан реального середовища можна вважати організаційними фактами, які можуть викликати або перешкоджати регуляторним або координаційним моделям. Представлення конкретного набору агентів, які наразі беруть участь у доступних групах (тобто агентів, які активно вирішили взяти на себе роль у певних групах організації), допомагає розібратися в контексті усвідомлення глобальних моделей координації та регулювання, що керують системою, організаціями зокрема. Крім того, організації мають чіткі уявлення про те, що відбувається у їх взаємопов'язаних структурах, в яких беруть участь агенти, про поточний стан організації, щоб агенти могли діяти відповідно.

**Пояснюваність** Навіть для пояснення, підхід МАОП дозволяє працювати з ним на різних рівнях, від мікрорівня до макро/глобального. Наприклад, завдяки чіткому представленню цілей, системи на основі агентів розроблені таким чином, що користувачі можуть пояснити конкретний вибір щодо дій, зроблених окремим агентом попиту, що має велике значення, коли ми маємо справу з автономними системами. Точно так само організації надають чітке представлення схем регулювання та координації, а також структур влади та комунікації, які можуть бути основою для пояснення глобальної структури колективних автономних систем, а також їх глобального функціонування.

**Інтеграція** штучного інтелекту є важливою особливістю для розвитку автономних систем у непередбачуваних та розвиваються середовищах. У цьому випадку MAOP забезпечує дисциплінований підхід до впровадження та використання методів ШІ. Зокрема, модель програмування агентів, прийнята в цій освітній дисципліні, базується на моделі/архітектурі BDI, а наукові внески в літературі описують розширення для інтеграції таких методів, як планування та посилення навчання. Тим не менш, ШІ можна також запровадити як послугу, загорнуту в ресурси середовища (артефакти), які агенти можуть використовувати для виконання конкретних завдань (наприклад, розпізнавання мовлення). Окрім індивідуального рівня, про планування та навчання можна подумати на рівні MAS, наприклад, впроваджуючи багатоагентське планування та багатоагентські стратегії навчання, описані в літературі.



У лекції 10 поданий огляд цих напрямків дослідження.

## Підсумок

У цьому розділі ми вперше представили багатоагентні системи як еталонну парадигму, яка використовується в цій книзі для моделювання та проектування програмних систем, а потім ми надали огляд ключових ідей щодо багатоагентного орієнтованого програмування, підходу до багатовимірного програмування для розробки MAS. Тоді ми повернулися до деяких основних викликів, які обговорювалися щодо того, які програми та системи підходять для MAS, обговорюючи загалом, наскільки ефективні ключові ідеї MAOP у вирішенні таких складностей. У решті дисципліни ми заглиблюємось у теорію та практику MAOP, починаючи з виміру агента (лекція 3). Перед цим у наступному розділі ми розпочинаємо роботу з JaCaMo, платформою, яка використовується для написання та запуску прикладів та програм, що з'являються у цій дисципліні.

## Джерела до лекції

Geraci, Anne, Freny Katki, Louise McMonegal, Bennett Meyer, John Lane, Paul Wilson, Jane Radatz, Mary Yee, Hugh Porteous, and Fredrick Springsteel. 1991. IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries. IEEE Press.

Morris, Edwin, Linda Levine, Craig Meyers, Pat Place, and Dan Plakosh. 2004. System of systems interoperability (SOSI): final report, Technical report, DTIC Document.

Tolk, Andreas, and James A Muguira. 2003. The levels of conceptual interoperability model. In Proceedings of the 2003 Fall Simulation Interoperability Workshop, Vol. 7, 1–11. Citeseer.

## Література

Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, Andrea Santi. Multi-agent oriented programming with JaCaMo. - Science of Computer Programming 78 (2013) 747–761.

Rafael C. Cardoso and Angelo Ferrando. A Review of Agent-Based Programming for Multi-Agent Systems. - Computers 2021, 10, 16. - <https://doi.org/10.3390/computers10020016> .

Multi-Agent Programming: Languages, Platforms and Applications. - © 2005 by Springer. - ISBN-10: 0-387-24568-5

Engineering Multi-Agent Systems. 6th International Workshop, EMAS 2018. Stockholm, Sweden, July 14–15, 2018. - © Springer Nature Switzerland AG 2019. - <https://doi.org/10.1007/978-3-030-25693-7>

Rafael H. Bordini · Mehdi Dastani · Jürgen Dix · Amal El Fallah Seghrouchni, Ed. Multi-Agent Programming: Languages, Tools and Applications. - © Science+Business Media, LLC 2009. - DOI 10.1007/978-0-387-89299-3

**Наступна лекція буде присвячена  
введенню до платформи JaCaMo.**