

ЛЕКЦІЯ № 6

ОСНОВИ ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ АГЕНТІВ ТА НЕЧІТКИХ МНОЖИН

ПЛАН

1. Основні властивості та архітектури програмних агентів
2. Мультиагентні системи
3. Основи теорії нечітких множин
4. Функції приналежності та операції над нечіткими множинами
5. Практичне застосування нечіткої логіки

ЛІТЕРАТУРА

Плескач В.Л., Рогушина Ю.В. Агентні технології: Монографія. – К.: Київ. нац. торг.–екон. ун–т, 2005. – 344 с.

ВСТУП

Програмні агенти (ПА) – досить нова парадигма програмування, яка дозволяє перейти на новий, більш інтелектуальний рівень взаємодії користувача з програмним і апаратним забезпеченням. Вона сприяє підвищенню ефективності праці та дозволяє користувачам доручити ІС виконання досить складних завдань.

Розподілене керування даними та обчислювальними процесами в умовах глобальних та корпоративних мереж спонукало до створення нової концепції середовища функціонування ПЗ як середовища взаємодії мультиагентних систем (МАС), кооперації і конкуренції інтелектуальних агентів.

1. Основні властивості та архітектури програмних агентів

Терміни «агент» і «інтелектуальний агент» (ІА) мають два значення, і через це іноді виникає плутанина.

У штучному інтелекті, під терміном **інтелектуальний агент** розуміють розумні сутності, що спостерігають за навколишнім середовищем і діють у ньому, при цьому їхня поведінка раціональна в тому розумінні, що вони здатні до розуміння і їхні дії завжди спрямовані на досягнення якої-небудь мети. Такий агент може бути як роботом, так і вбудованою програмною системою. Про інтелектуальність агента можна говорити, якщо він взаємодіє з навколишнім середовищем приблизно так само, як діяла би людина.

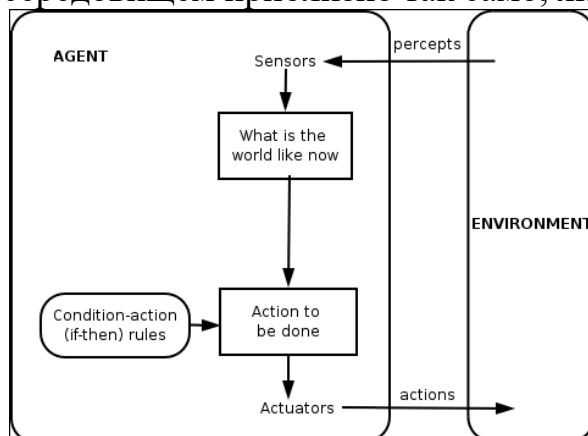


Рис. 1. Звичайний агент

У комп'ютерній науці, інтелектуальний агент — це програма, що самостійно виконує завдання, указане користувачем комп'ютера, протягом тривалих проміжків часу. Інтелектуальні агенти використовуються для сприяння операторові або для збирання інформації. Одним із прикладів завдань, виконуваних агентами, може служити завдання постійного пошуку й збору необхідної інформації в Інтернеті. Комп'ютерні віруси, боти, пошукові роботи — усе це також можна віднести до інтелектуальних агентів. Хоча такі агенти мають строгий алгоритм, «інтелектуальність» у цьому контексті розуміється як здатність пристосовуватися й навчатися.

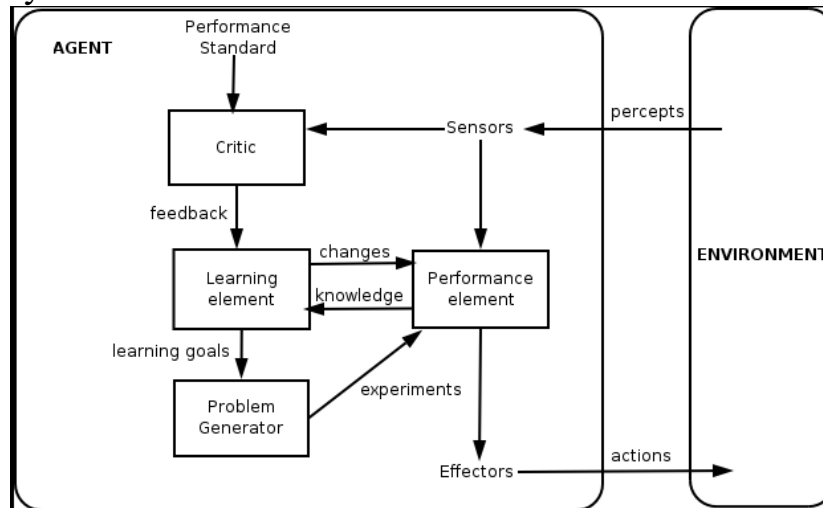


Рис. 2. Інтелектуальний агент

Для програмістів Агент – це, насамперед, комп'ютерна програма. З цього випливають такі властивості, як коректність, повнота, ефективність, надійність. При цьому агент виконує певні функції людини, надаючи користувачу потрібні йому послуги.

Можна визначати ІА через **множину його атрибутів**. ІА – термін, що дозволяє об'єднати множину більш специфічних і обмежених типів агентів, які мають деякі з таких атрибутів:

- реактивність (reactivity) – зміна своєї поведінки залежно від конкретної ситуації;
- автономність (autonomy) – самостійне виконання розпоряджень користувача без детальних інструкцій;
- співробітництво (collaborative behavior) – здатність працювати разом з іншими агентами для досягнення спільної мети;
- спілкування на рівні знань (“knowledge level” communication ability) – спроможність спілкуватися з людьми й іншими агентами мовою, близькою до природної;
- здатність до логічного виведення (inferential capability) – обробка абстрактного опису задачі з використанням апріорних знань про цілі і найбільш придатні методи їх досягнення, спроможність будувати моделі власної сутності, свого користувача, ситуацій та інших агентів;
- безперервність у часі (temporal continuity) – стійкість ідентифікації і положення протягом тривалого періоду часу;
- персоналізація (personality) – наявність персоналізованих значень атрибутів власної поведінки;
- адаптивність (adaptivity) – навчання й удосконалення на основі власного досвіду;
- мобільність (mobility) – здатність самостійно переходити з однієї платформи на іншу.
- правдивість – припущення про те, що агент не буде свідомо поширювати помилкову інформацію;

- лояльність – намагання робити те, що потрібно іншим агентам;
- раціональність – виконання тільки тих дій, які приводять до досягнення цілей.

Класифікація ПА за призначенням:

- інтерфейсні агенти;
- Інтранет-агенти;
- Інтернет-агенти;
- гетерогенні агенти.

Інтерфейсні агенти можна розглядати як персональні асистенти, які допомагають користувачу працювати з різними програмними засобами. Такі агенти спостерігають за діями користувача та намагаються запропонувати йому такі дії, які мають спростити його роботу. Вони здатні адаптуватися до індивідуальних особливостей та потреб конкретного користувача. *Інтерфейсні ПА* здатні до самонавчання, що приводить до ефективнішого виконання ними своїх функцій. Для самонавчання агенти використовують:

- спостереження за поведінкою користувача;
- отримання позитивних або негативних оцінок від користувача (зворотний зв'язок);
- безпосереднє отримання інструкцій від користувача;
- консультації з іншими агентами.

Представниками цієї групи агентів є такі системи, як A-Match та Verbal Software Robots.

Інтернет-агенти є найбільш численними представниками інтелектуальних ПА. Вони виникли як засіб обробки та транспортування інформаційних ресурсів Інтернету. На відміну від інтерфейсних агентів вони здатні не тільки створювати персоніфікований профіль користувача, але й відповідно класифікувати інформаційні ресурси.

Інформаційний ресурс (ІР) – це будь-яка інформація, що має цінність у певній предметній області (ПрО) та може використовуватися людиною (явно або через ПЗ) для досягнення певної мети. Переважна більшість ІР зараз може розглядатися як документи, проте існують й інші види ІР.

Інтернет-агентів можна поділити на дві основні групи: *статичні та мобільні*. Статичні ПА звичайно вбудовані у браузер. Приміром, такі агенти можуть сортувати електронну пошту, повідомляти користувача про події та повідомлення, які, за наявними в агента відомостями, можуть його зацікавити. Мобільні Інтернет-агенти менш поширені. Прикладом такого агента є Jasper. Цей агент здатний не тільки знаходити інформацію, цікаву для його користувача, але й повідомляти про неї інших агентів.

Робота *Інтранет-агентів* схожа на роботу Інтернет-агентів, проте має власну специфіку. Характерні задачі, завдяки яким Інтранет-агентів виділяють у окрему групу, такі:

- автоматизація бізнес-процесів підприємства;
- виконання послуг для користувачів, пов'язаних з використанням інформації з бази даних підприємства.

Гетерогенні агенти інтегрують функції двох або більше агентів, які належать до різних типів.

На відміну від класичних систем ШІ агенти **не тільки пропонують рішення проблеми, але й реально діють**. Сукупність причин, через які ІА виконує певні дії, називають **мотивацією**. Для досягнення своїх цілей ІА конструює план дій, які мають призвести до виконання поставленого перед ним завдання.

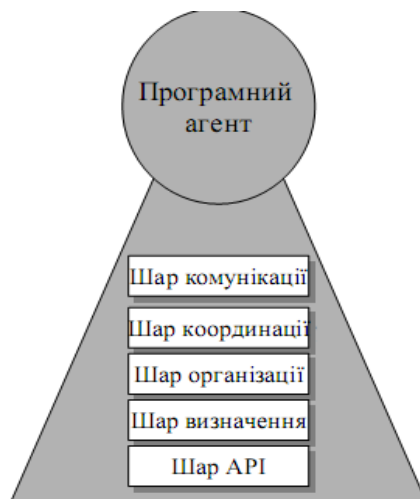


Рис. 3. Багатошарова схема ІА

ПА функціонують у багатовимірному просторі. Вони складаються з кількох шарів (рис.3):

- комунікації;
- координації;
- організації;
- визначення;
- інтерфейсу API.

Шар комунікації розглядає низькорівневі деталі взаємодії між ПА. На *координаційному шарі* подаються соціальні властивості ПА, технології координації та переговорів. На *організаційному шарі* ПА визначається через відношення з іншими ПА, через ролі, які він виконує у взаємодії з цими агентами. На *шарі визначення* ПА визначається як автономна раціональна сутність, тобто в термінах механізмів міркування та навчання, цілей, ресурсів, здібностей, переконань тощо. Шар *програмного інтерфейсу API* пов'язує ПА з його фізичною реалізацією.

В обчислювальному середовищі кожний агент має свій життєвий цикл і власне ім'я. За допомогою використання доменів призначення агентів відбувається їх адміністрування.

Позначення імені агента забезпечує спосіб ідентифікації агента серед доменів інших груп агентів. Розробка програмних агентів вимагає застосування стандартизованих профілів агентів і методології розробки агентів для кожного з конкретних застосувань.

З погляду користувача, основна перевага використання агентів полягає в спрощенні взаємодії з програмою – користувачу досить поставити загальну задачу перед агентом, не вдаючись у подробиці того, як саме агент має її вирішувати. Якщо агент має можливості для рішення цієї задачі, він вирішує її сам, а інакше запитує необхідні йому послуги в інших агентів.

Для користувача *Агент* – це програмний об'єкт, який:

- забезпечує виконання однієї або кількох корисних послуг;
- надає опис семантики цих послуг іншим ПА;
- здатний функціонувати автономно без безпосередніх вказівок користувача;
- може інтерактивно взаємодіяти з іншими ПА та користувачами.

Функціональний зв'язок між агентом, середовищем, у якому цей агент функціонує, та його програмним кодом можна подати в такий спосіб: $A=f(X)$, де

- A – агент;
- f – програма (функція);
- X – середовище.

Агент може описуватися як функція f, що обробляє інформацію від сенсорів і вхідні повідомлення (рис.4). Результат роботи агента – дії і вихідні повідомлення. Цей узагальнений підхід відповідає як біологічним, так і інтенціональним моделям агентів. Розходження між цими моделями полягають у способі визначення f.

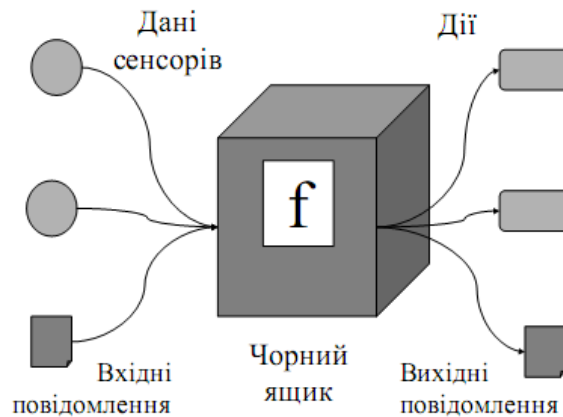


Рис. 4. Визначення агента через модель чорного ящика

Однією з основних характеристик агента є комунікабельність – здатність до гнучкого спілкування як з агентами, так і з іншими програмними компонентами. Унаслідок цього агенти відіграють важливу роль у досягненні *інтероперабельності* ПЗ, створеного незалежними розробниками в різний час [64]. Агентифікація дозволяє поширити цю властивість на довільне ПЗ. Агентифікація – перетворення довільного програмного забезпечення в ПА, приміром, у формі надбудови агентної оболонки над фрагментами програмного коду, яка забезпечує інтероперабельність цього коду [212].

Підкласом ПА є інтелектуальні агенти (ІА) (рис.2.3).

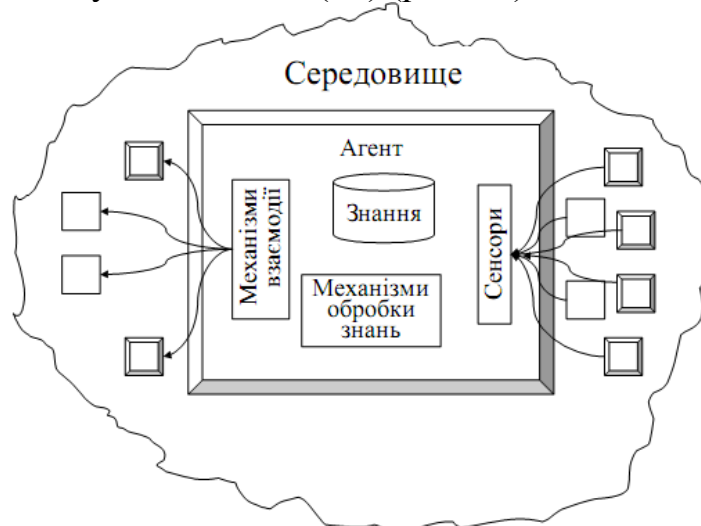


Рис. 5. Загальна концептуальна схема інтелектуального ПА

Прийнято розрізняти вузьке ("сильне") та широке ("слабке") визначення терміну "інтелектуальний агент".

ІА у широкому розумінні – це ІС, що має такі ключові ознаки:

- автономність (autonomy) – функціонування значною мірою незалежно від втручання людини і контроль власних дій та внутрішнього стану;

□ соціальність (social ability) – інтелектуальна та конструктивна взаємодія з іншими агентами та людьми шляхом обміну з ними повідомленнями деякою загальнозрозумілою мовою комунікацій;

□ реактивність (reactivity) – сприйняття зміни середовища і вчасне реагування на них;

□ проактивність (pro-activity) – здатність агента генерувати цілі і діяти раціонально для їх досягнення, а не тільки реагувати на зовнішні події.

Зазвичай, поведінка людини прогнозується і аналізується через такі атрибути відношень, як переконання, бажання, надії, побоювання тощо, які називаються інтенціональними поняттями.

Більш строге ("сильне") розуміння терміну "інтелектуальний агент" вимагає наявності в агента ментальних властивостей (інтенціональних відношень), до яких належать:

□ знання (knowledge) – стала частина інформації агента про себе, середовище й інших агентів, що не змінюється в процесі його функціонування;

□ переконання (beliefs) – знання агента, які можуть змінюватися в процесі його функціонування і ставати хибними, про поточний стан світу і про зміни в ньому, до яких має привести виконання дій агента;

□ бажання (desires) – ставлення агента до майбутніх станів світу та переваги, які він надає одним з них порівняно з іншими (агент може мати несумісні та недосяжні бажання і тому не очікує, що усі вони мають бути досягнуті);

□ наміри (intentions) – підмножина цілей, які може досягти обмежений у ресурсах агент, і засіб їх досягнення;

□ цілі (goals) – несуперечлива підмножина бажань, досягнення яких агент прийняв як поточну стратегію поведінки;

□ зобов'язання (commitments) стосовно інших агентів – завдання, що агент виконує за дорученням інших агентів у рамках кооперації та співробітництва.

Перші два поняття – переконання та знання – називають "точкою зору" (attitudes) агента, інші характеризують в англійській літературі загальним терміном "pro-attitude", сутність яких полягає у тому, що вони спрямовують дії та поведінку агента.

Інтенціональні відношення поділяються на інформаційні (переконання та знання) і перед-відношення (бажання та емоції, намір, зобов'язання, цілі тощо). Перші стосуються інформації, що має агент про світ, у якому він існує, тоді як передвідношення – це відомості, які певним чином впливають на дії агента.

Перед- і інформаційні відношення тісно пов'язані, оскільки агенти можуть, наприклад, формувати наміри на основі наявної в них інформації про світ.

Деякі автори вважають, що інтелектуальному агенту мають бути притаманні такі властивості:

□ мобільність (mobility) – здатність переміщуватися телекомунікаційними мережами (локальними або глобальними) для досягнення своїх цілей;

□ доброзичливість (benevolence) – готовність агентів допомагати іншим агентам та виконувати доручення користувача;

□ правдивість (veracity) – властивість не повідомляти іншим агентам та користувачу інформацію, про помилковість якої йому відомо;

□ раціональність (rationality) – здатність виконувати саме ті дії, що приводять до досягнення його цілей у рамках наявних у агента знань і переконань.

Інтелектуальність ПА визначається його спроможністю міркувати і навчатися, наявністю моделі користувача, його потреб і механізму пошуку засобів їх задоволення.

Модель ПА містить модель ПрО, модель користувача, засоби сприйняття (сенсори), засоби виконання дій (ефектори), цілі і планувальник дій на підставі цілей, моделі інших агентів і засоби взаємодії з ними

Таксономії програмних агентів

Для опису агентів більш ефективним способом, порівняно з аналізом усіх можливих комбінацій атрибутів, є різні схеми і таксономії. Приміром, Гілберт описує ПА в термінах тривимірного простору через дії, інтелект та мобільність (рис.6).

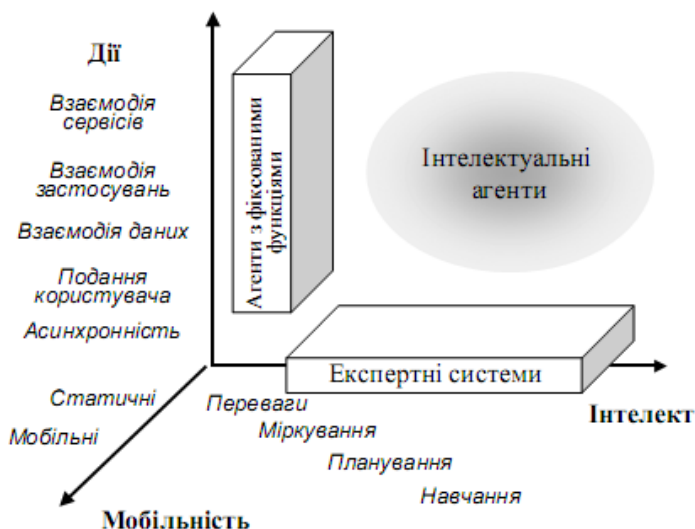


Рис.6. Класифікація агентів Гілбертом у термінах тривимірного простору дій, інтелекту та мобільності

Дамо визначення цих базових термінів за Гілбертом.

Дія – рівень автономності і повноважень агента, що може оцінюватися через природу взаємодії між агентом і іншими сутностями в системі. Агент має бути асинхронним. Його ступінь дії підвищується, якщо агент має певне уявлення про користувача. Більш розвинутий агент може взаємодіяти з даними, застосуваннями й іншими агентами.

Інтелект – рівень поведінки, пов'язаний з міркуваннями і навчанням. Здатність агента сприймати твердження користувача щодо цілей і виконувати його завдання пов'язана саме з цією якістю.

Як мінімум, агент має вміти визначати пріоритет тверджень. Вищий рівень інтелекту потребує моделі користувача і здатність до міркувань. Ще більш інтелектуальними є системи, здатні навчатися за власним досвідом для того, щоб адаптуватися до свого середовища.

Мобільність – рівень самостійності агента при його переміщеннях мережею. Мобільні сценарії можуть формуватися на одній машині і надсилати запит для виконання на інший вузол мережі. Мобільні об'єкти переміщуються між вузлами під час виконання, переносячи накопичені ними дані.

Таксономія агентів, яку пропонують Франклін та Грассер [81], дозволяє описати більшість відомих прикладів ПА. Ця таксономія поділяє агентів за структурами керування, середовищем функціонування (приміром, база даних, файлова система, мережа Internet), за мовою, якою вони написані, і за застосуванням (рис.7).

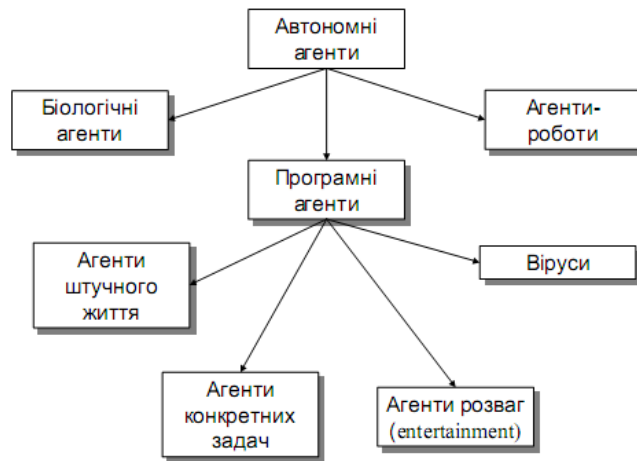


Рис. 7. Таксономія агентів Франкліна та Грассера

Архітектури агентів

Основні принципи побудови та функціонування разом з структурою ПА називають їх архітектурою. Залежно від того, які принципи визначають дії агентів, архітектури поділяються на **деліберативні** (агенти обирають план дії на основі логічного виведення з наявних в них знань) та **реактивні** (дії агентів визначаються як реакція на події у зовнішньому середовищі).

На практиці зазвичай застосовують різноманітні комбінації цих архітектур, які називають **гібридними**. Крім того, деякі дослідники виділяють в окремі класи архітектури з певними спільними рисами (приміром, інтерактивні архітектури, архітектури з планувальником дій, архітектури інтелектуальних агентів, інтенціональні архітектури).

Архітектура ПА відображає внутрішню організацію та взаємодію між основними компонентами (рис.8).

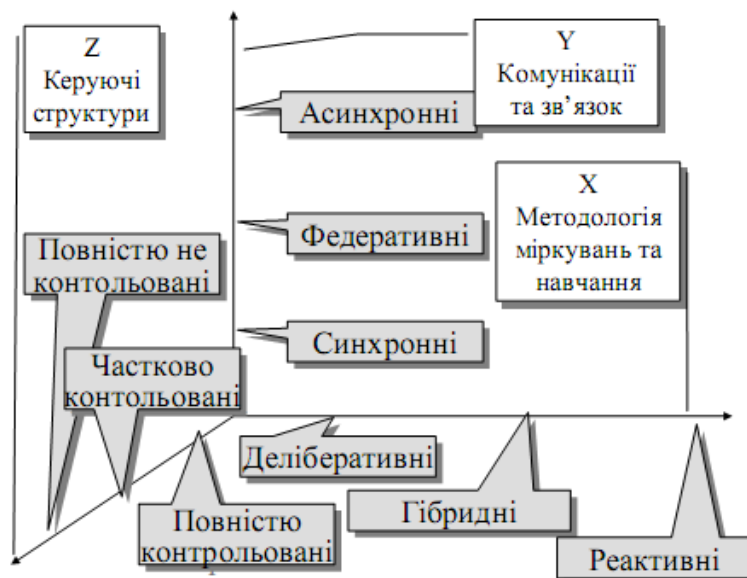


Рис.8. Таксономія архітектур мобільних агентів

Деліберативна архітектура ПА містить символічну модель світу, подану у явній формі, за допомогою якої ПА на основі міркувань логічного чи псевдологічного типу приймає рішення про дії, які він здійснює. Такий агент може розглядатися як спеціальний випадок системи, заснованої на знаннях. Використання деліберативної архітектури ПА приводить до кількох *принципових проблем*:

- обсяг символної інформації, яку зберігає ПА, пропорційний розміру внутрішнього стану, тому його збільшення призводить до зниження мобільності ПА;
- символне подання інформації про середовище, на основі якої міркують ПА, має формуватися у режимі реального часу, щоб знайдені агентами рішення були корисними;
- перетворення сценаріїв реального світу на точне та адекватне символне визначення є нетривіальною задачею.

Реактивні архітектури не використовують централізовану символну модель світу та не застосовують складні символні міркування. Така архітектура оперує на низькому рівні абстракції.

Невеликий час очікування відповіді забезпечує ефективну взаємодію між агентами з такою архітектурою один з одним та з середовищем.

Недолік архітектури – неможливість глибокого аналізу даних від сенсорів.

Реактивна архітектура забезпечує прийняття рішень за значно меншим обсягом інформації про оточуюче середовище та за значно менший час, використовуючи прості емпіричні правила, специфічні для певної Про.

З іншого боку, архітектури агентів класифікують залежно від типу структури функціональних компонентів ПА та методів організації взаємодії між ними. *Як правило, архітектура агента організується у вигляді кількох рівнів.*

Тільки найпростіші ПА можуть бути реалізовані за однорівневою схемою. Рівні відображають функції ПА [277], такі, як сприйняття зовнішніх подій і прості реакції на них; координація взаємодії з іншими ПА; відновлення переконань про зовнішній світ; визначення своїх дій на черговому кроці тощо. Найчастіше в архітектурі ПА присутні рівні, що відповідають за такі функції:

- сприйняття і виконання дій;
- реактивну поведінку;
- локальне планування;
- кооперативну поведінку;
- моделювання Про;
- формування намірів;
- навчання.

Існує два класи багаторівневих архітектур агентів :

- горизонтальна модульна;
- вертикальна модульна.

Залежно від моделі подання знань використовується у ПА виділяють:

- конекційну архітектуру (з використанням нейронної мережі);
- архітектуру, що базується на правилах.

Залежно від того, які технології ШІ реалізовано в ПА, виділяють архітектуру, що базується, на:

- генетичних алгоритмах;
- знаннях.

Горизонтальна модульна архітектура (рис.9) є однією з найпоширеніших. У такій архітектурі функції агента подані окремими модулями у загальному вигляді, без специфікації за задачами.

Між основними блоками існують функціональні зв'язки. Залежно від організації агента вхідна інформація реєструється безпосередньо у БД агента, використовується для навчання агента або для отримання завдань користувача.

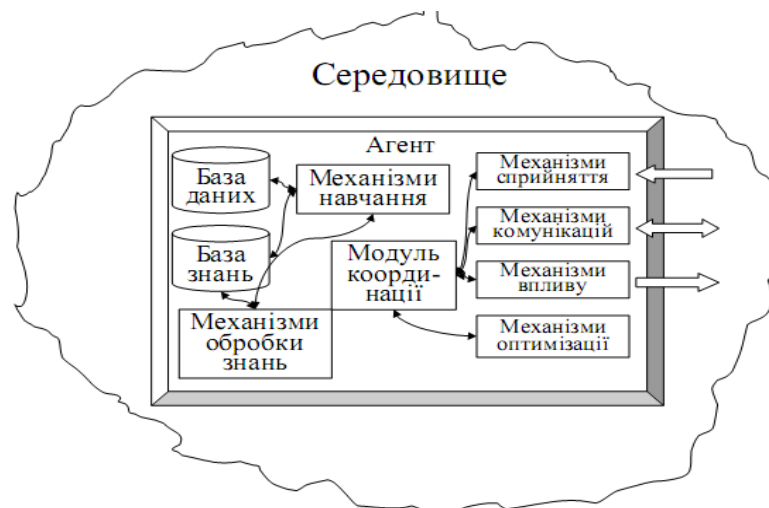


Рис.9. Горизонтальна модульна архітектура ПА

Така архітектура характерніша для "розмірковуючих" агентів, а не для реагуючих.

Вертикальна модульна архітектура виділяє модулі агента, які відповідають певним аспектам взаємодії ПА з зовнішнім середовищем. Така архітектура характерніша для реагуючих агентів.

Функціональні зв'язки між модулями є ієрархічними. Модулі виконуються паралельно. Якщо два модулі вступають у конфлікт, розглядаються дані, які поступили від домінантного модуля.

Конекційна архітектура ПА базується на конекційній моделі подання знань, яка використовує багатошарові (multilayer) нейронні мережі. Сигнал, що виходить з одного нейрона, може бути вхідним для іншого. Нейронні мережі використовуються в архітектурі ПА за такими причинами:

- у простих випадках зв'язки встановлюються безпосередньо;
- нейронна мережа може сама встановлювати зв'язки між нейронами через механізм "back-propagation". Різниця між очікуваними та отриманими результатами впливає на встановлення зв'язків між шарами мережі (зворотний зв'язок);
- зв'язки встановлюються за допомогою генетичних алгоритмів.

Архітектура, що базується на правилах (рис.10), розглядає ПА як продукційну систему, здатну сприймати середовище та впливати на нього. Сигнали, що поступають від зовнішнього середовища, фіксуються в базі даних ПА. Одночасно з цим працює механізм виведення, який обирає та виконує правила, що відповідають поточному стану БД.

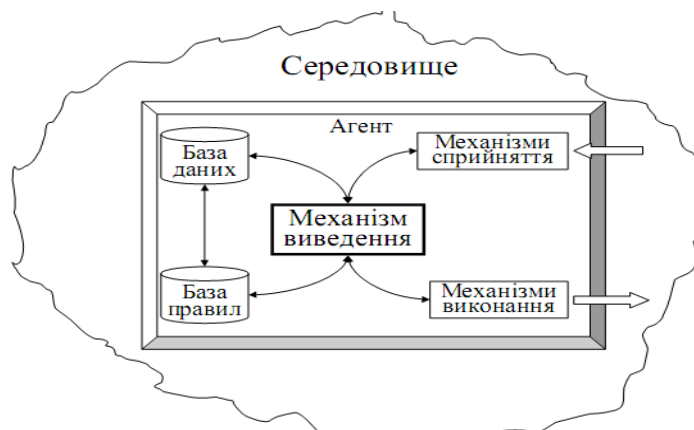


Рис.10. Архітектура ПА, що базується на правилах

Архітектура типу "чорна дошка" застосовує поділ на незалежні модулі знань, які взаємодіють один з одним не безпосередньо, а через розподілені дані (рис.11). Модулі працюють у просторі, який називають "чорною дошкою". Він містить елементи, які необхідні для вирішення проблем взаємодії. "Чорна дошка" складається з ієрархічно пов'язаних підпросторів: гіпотез, проміжних результатів і різноманітних даних, якими обмінюються модулі. Контрольний механізм керує конфліктами між модулями.

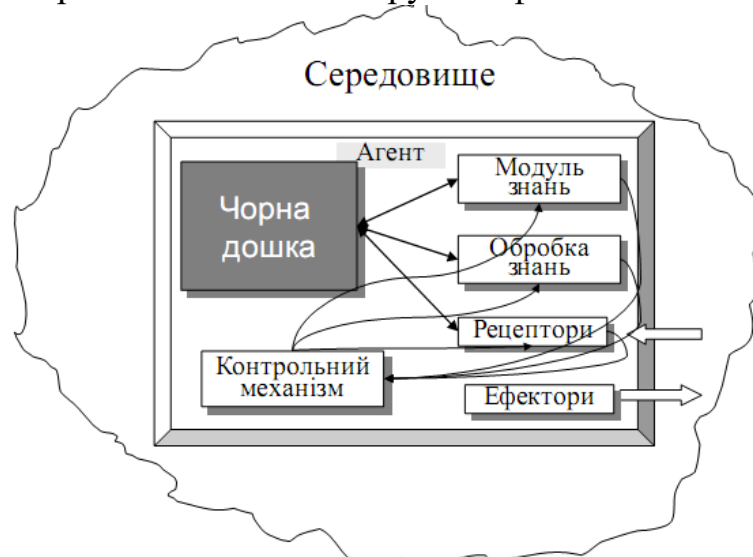


Рис.11. Архітектура ПА типу "чорна дошка"

Архітектура на основі генетичного алгоритму (рис.2.10) дозволяє агентам успадковувати властивості двох батьків-агентів.

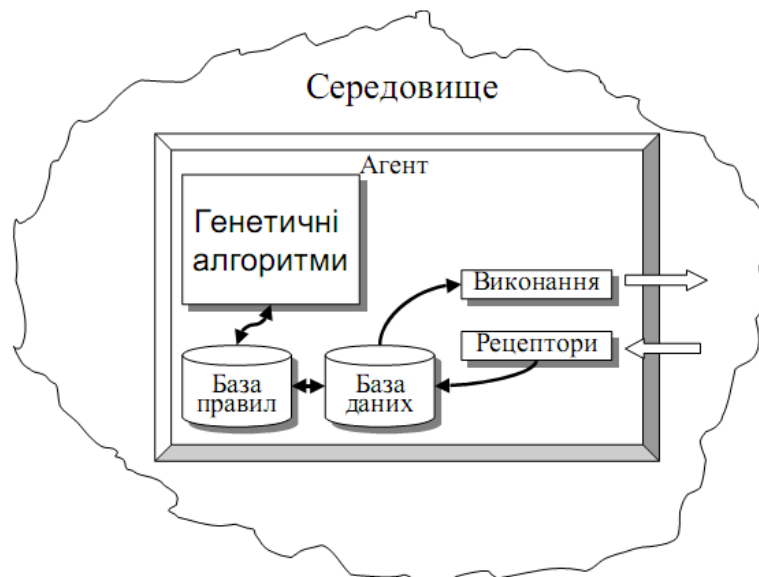


Рис.2.10. Архітектура ПА на базі генетичного алгоритму

2. Мультиагентні системи

Проблеми, для вирішення яких застосовують агентний підхід, можуть бути досить складними. Дослідження в галузі ШІ показали, що взаємодія і декомпозиція загальної задачі ефективно позначається на результатах виконання цієї задачі. Тому краще створювати окремі автономні блоки такої системи, а потім організувати їх спільне функціонування.

Термін "мультиагентні системи" використовується для позначення ІС, які складаються з множини автономних модулів ПА та мають такі **властивості**:

- кожен ПА є автономним, мобільним та інтероперабельним;
- ПА, що входять до складу МАС, здатні обмінюватися інформацією для досягнення спільних цілей;
- керування ПА може бути децентралізованим;
- джерела даних і доступ до них децентралізовані;
- робота агентів є асинхронною.

Теорія МАС походить від теорії відкритих систем, розподіленого ШІ і загальної теорії складних систем.

Розподілений штучний інтелект (РШІ) пов'язаний з аналізом систем, що складаються з окремих незалежних об'єктів, які взаємодіють один з одним, та механізмів їх координації. МАС теж є предметом розгляду РШІ. У цьому випадку незалежними об'єктами є ПА. Для вивчення поведінки МАС використовують методи таких наукових дисциплін:

- розподілений ШІ (теорія розподілених систем, теорії прийняття рішень), що займається найбільш загальними аспектами колективної поведінки агентів;
- теорія ігор, яка використовується для дослідження ситуацій, аналогічних до кооперативних ігор, стратегій ведення переговорів;
- теорія колективної поведінки автоматів, яка досліджує колективну поведінку великих груп автоматів з примітивними функціями, спроможних навчатися за допомогою системи штрафів і заохочень;
- біологічні, економічні та соціальні моделі.

У розробці МАС використовуються також результати з інших галузей теоретичних досліджень (рис.13).

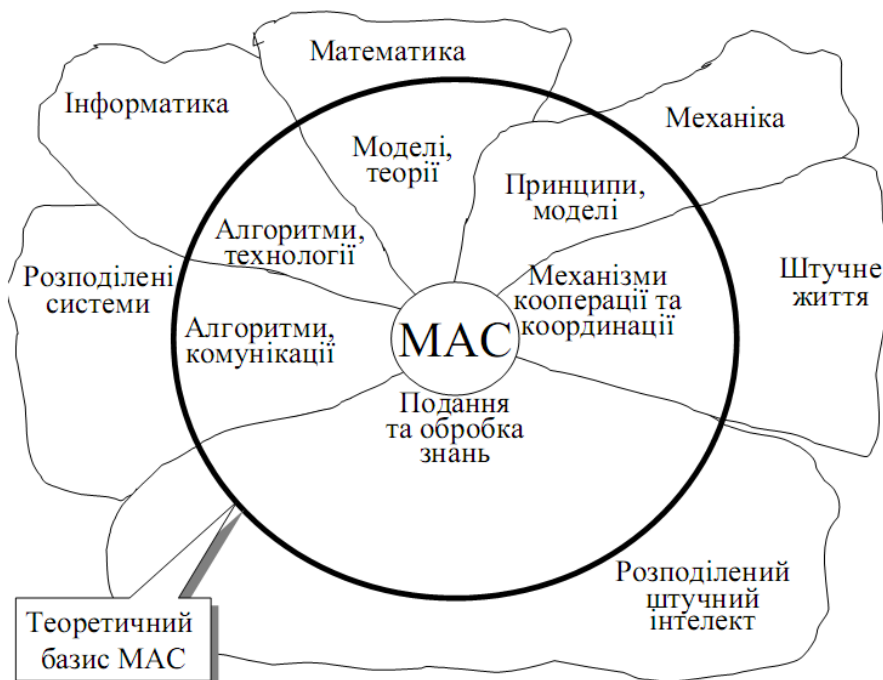


Рис.13. Основні Про, що використовуються у розробці МАС

МАС – співтовариство ПА, які пов'язані один з одним цілями та ресурсами. ПА, що входять до складу МАС, здатні взаємодіяти для того, щоб обмінюватися послугами, які потрібні їм для досягнення цілей, поставлених перед ними користувачами. Агент, не

здатний самостійно вирішити задачу, поставлену перед ним користувачем, звертається до інших агентів, які можуть надати йому відповідні послуги.

МАС поширені в багатьох ПрО: керуванні виробничими процесами і промисловими підприємствами; плануванні рухом транспорту (повітряного, залізничного, автомобільного); аналізі та пошуку інформації; навчанні; бізнесі тощо.

МАС складається з множини ПА, які:

- взаємодіють шляхом комунікацій;
- здатні діяти у певному середовищі;
- мають певні сфери впливу, які можуть перетинатися або співпадати.

Можна розглядати МАС як слабкопов'язану мережу обчислювальних пристроїв, взаємодіючих для вирішення проблеми, що знаходиться поза індивідуальними можливостями або знаннями кожного з цих пристроїв.

Координація – одна з центральних проблем МАС. Наступні фактори визначають потребу в координації МАС:

- запобігання хаосу в децентралізованій МАС;
- наявність глобальних обмежень, яким має задовольняти МАС для успішного виконання завдання;
- розподіленість знань, ресурсів та інформації, які використовують ПА, що входять до складу МАС;
- залежність між цілями та діями ПА;
- забезпечення ефективності функціонування МАС.

Функціонування МАС пов'язане з кооперацією та конкуренцією агентів в процесі колективного вирішення задач. Агент, що не може вирішити власну задачу самостійно, має взаємодіяти з іншими ПА.

При цьому агенти можуть будувати плани спільних дій, ґрунтуючись не тільки на власних можливостях, але і аналізувати плани і наміри інших ПА (використовуючи різні комбінації інтенціональних відношень). Кооперативні дії ПА – важлива перевага МАС. При такій організації роботи група ПА проявляє новий, ефективніший тип поведінки.

Мотивація використання МАС базується на їх властивостях:

- здатність вирішувати проблеми, складні для одного централізованого ПА через обмеженість ресурсів;
- можливість взаємодії та інтеперабельності з різноманітними застосунками (ЕС, СППР тощо);
- вирішення дійсно розподілених проблем (приміром, керування рухом транспорту);
- знаходження рішень на основі розподілених ІР;
- вирішення проблем з розподіленою експертизою (приміром, медичний догляд);
- підвищення швидкості та надійності;
- здатність збільшувати кількість обчислювальних пристроїв (процесорів), що використовуються для виконання задачі;
- можливість опрацювання нечітких та неповних даних і знань.

Знання й уміння МАС здобуваються від великої кількості відносно простих ПА, що поєднуються разом деякою архітектурою. В процесі колективної роботи агенти мають будувати плани дій, ґрунтуючись не тільки на своїх можливостях, але й враховуючи плани і наміри інших агентів (на основі своїх знань та переконань щодо цих планів та намірів).

Причини взаємодії ПА:

- наявність сумісних цілей;
- нестача індивідуальних ресурсів ПА для досягнення цілей;
- нездатність ПА самостійно вирішити задачу;
- наявність взаємних зобов'язань.

Перехід до відкритих МАС надає можливість переходу на нову якість функціонування системи в силу того, що система (група агентів) більше, ніж сума властивостей її членів (агентів) **ГЕШТАЛЬДТ**.

Системи, які складаються з досить простих програм, кожна з яких переслідує тільки свої примітивні цілі, у цілому здатні вирішувати дуже складні задачі. *Однак моделювання колективної поведінки приводить до необхідності рішення багатьох проблем:*

- формування спільних планів дій;
- врахування інтересів інших агентів;
- синхронізації спільних дій;
- вирішення конфліктів між цілями різних агентів;
- конкуренції за спільні ресурси;
- організації переговорів про спільні дії;
- розпізнавання необхідності кооперації;
- вибір партнерів;
- декомпозиції задач;
- поділу обов'язків тощо.

Багато робіт в сфері ШІ присвячені специфікації протоколів взаємодії між ПА. Дослідження переговорів у МАС засновані на ідеї *контрактних мереж*. Відповідно до цієї парадигми ПА, який бажає отримати певну послугу, запрошує на переговори інших ПА і вибирає послуги, які найбільше відповідають його потребі. У складніших моделях ПА тільки запитує пропозиції, а інші спеціалізовані агенти оцінюють варіанти відповідей та обирають найкращу.

Інший вид взаємодії між ПА – *формування коаліції* для спільного виконання своїх намірів. Механізми формування коаліції, де агенти динамічно поєднуються з іншими агентами, інтенсивно вивчаються багатьма дослідниками.

Існує багато моделей кооперації агентів. Приміром, модель CPS (Cooperative Problem Solving) призначена для встановлення взаємодії між ПА, побудованих на основі VDI-архітектури. Ментальні поняття формалізуються за допомогою операторів темпоральної логіки для визначення таких понять, як потенціал кооперації, групові дії, досяжність мети агента тощо.

Найпростіший метод координації МАС – *організаційне структурування* – полягає у чітко визначених і довгострокових відношеннях між ПА. При цьому використовують ієрархічні структури master-slave або client-server. Організаційне структурування передбачає, що принаймні один ПА має глобальне уявлення про все співтовариство, однак для багатьох ПрО це не реально.

Цей метод використовується в двох варіантах:

- головний ПА планує і розподіляє завдання між підлеглими ПА, які, на відміну від головного, мають часткову автономність;
- для координації використовується дошка оголошень для обміну інформацією між ПА, операції з якою (зчитування й запис) визначає головний ПА.

Вузким місцем МАС, які базуються на технології дошки без прямої взаємодії ПА, є збільшення кількості агентів у співтоваристві та необхідність спільного представлення

про дошку. Тому більшість систем, заснованих на цій технології, використовують невеликі гомогенні ПА (наприклад, DVMT).

Інший варіант координації – *укладення контракту* – припускає децентралізовану структуру. ПА мають дві ролі:

- менеджер поділяє задачу на підзадачі і шукає виконавця, щоб виконати їх;
- виконавець реалізує свою підзадачу.

Виконавець може рекурсивно стати менеджером і розбити свою підзадачу на ще дрібніші задачі і доручити їх іншим ПА.

Пошук виконавця складається з таких етапів:

- менеджер повідомляє задачу;
- виконавці оцінюють цю задачу щодо можливості її виконати;
- менеджер отримує таблицю виконавців, оцінює отримані пропозиції, обирає виконавця і доручає йому виконання певної задачі;
- виконавець виконує отриману задачу та повідомляє менеджера про отримані результати.

Така модель координації забезпечує розподіл задачі і засоби для самоорганізації співтовариства агентів.

Переваги цього підходу координації полягають у динамічному розподілі задачі завдяки пошуку оптимального виконавця;

збалансованому завантаженні ПА і надійному механізмі для розподіленого керування.

Існує два типи планування діяльності МАС:

- централізоване;
- розподілене.

У *централізованому плануванні* діяльності МАС є координаційний ПА, що отримує індивідуальні плани від інших, аналізує їх, щоб знайти потенційні протиріччя та конфлікти у взаємодії ПА. Потім координаційний ПА намагається змінити ці індивідуальні плани і поєднає їх у план МАС, у якому усунені суперечливі взаємодії та додані команди зв'язку, що синхронізують взаємодію ПА у потенційно можливих конфліктах.

У *розподіленому плануванні* діяльності МАС кожен ПА отримує моделі планів інших ПА, що входять до складу МАС. ПА взаємодіють, щоб побудувати модифікації своїх індивідуальних планів, які не конфліктують з планами інших ПА. Координація в розподіленому плануванні діяльності МАС набагато складніша, ніж у централізованому випадку, тому що жоден ПА не володіє глобальним уявленням про всю розподілену систему.

Переговори – один з найскладніших методів координації, який використовує методи ШІ, логіку тощо. У динамічному співтоваристві агенти самостійно розподіляють роботи між собою. Часто одну і ту ж роботу в співтоваристві можуть виконати кілька ПА. Агенти на момент розподілу робіт знаходяться в різних станах (приміром, мають різний ступінь завантаженості). Для ефективного розподілу робіт між ПА вони мають взаємодіяти один з одним – вести переговори, у процесі яких виявити оптимального виконавця для кожної роботи .

Процес переговорів складається з таких компонентів:

- протоколу переговорів – набору правил, за якими відбувається взаємодія агентів (учасники переговорів, їх типи, стани переговорів, правила, за якими змінюються стани переговорів, можливі дії учасників тощо);
- об'єкта переговорів – діапазону проблем, відносно яких потрібно досягти згоди;

□ моделі ухвалення рішення ПА – апарат ухвалення рішення, що використовують учасники відповідно до протоколу переговорів.

У процесі переговорів агент-ініціатор переговорів шукає потенційного агента-виконавця для виконання деякої роботи. З огляду на раціональність агентів розумним було б припустити, що об'єктом переговорів агентів на стадії розподілу робіт є розмір винагороди, яку одержує агент-виконавець за ці роботи.

У процесі формування кооперативного рішення в CPS-моделі виділяють чотири етапи:

1. Визначення потреби у кооперації (приміром, агент має певну мету, але переконаний у тому, що не може досягти самостійно).

2. Створення групи агентів. При успішному завершенні цього етапу створюється група ПА зі спільними зобов'язаннями, пов'язаними з виконанням колективних дій.

3. Формування спільного плану дій. ПА ведуть переговори для формування плану дій, який за їхніми переконаннями має привести до реалізації цілей кожного з них.

4. Спільні дії. Агенти виконують дії відповідно до плану, який вони прийняли на попередньому етапі, виконуючи прийняті на себе зобов'язанням.

Кожний з підходів до координації МАС має певні переваги і недоліки та може застосовуватися тільки для деяких специфічних областей. Універсального методу координації, що ідеально підходить для будь-якої реальної задачі, не існує.

Отже, при побудові МАС, що вирішує реальну задачу, як правило, необхідно використовувати комбінацію описаних вище підходів.

Основні типи конфліктів у МАС :

□ в системі переконань ПА, що можуть виникнути внаслідок отримання від іншого ПА інформації, що є хибною або суперечить його переконанням;

□ обумовлені неповнотою моделі середовища і моделей інших ПА;

□ внаслідок конкуренції ПА за спільні ресурси або через суперечливість їх цілей.

Вирішення конфлікту – це зняття логічного протиріччя шляхом відкидання однієї або обох альтернатив відповідно до певного критерію. Існує багато різних механізмів вирішення конфліктів:

централізовані, імовірнісні, відповідно до певних правил (на основі пріоритетів переконань, рівнів компетентності ПА) тощо.

Значна частина МАС, створених на сьогодні, складається з ПА, які не є повністю автономними (тобто здатними вибирати для себе цілі і вирішувати, яким чином досягати ці цілі та які саме дії виконувати для цього). Набір ПА розробляється цілісно, і проблема взаємодій між агентами вирішується в процесі їхньої розробки. Такі проблеми, як конфлікти або недостатність ресурсів, узагалі не розглядаються. Агенти, що входять до складу такої МАС, мають обмежену автономію: їхня роль у процесі рішення загальної проблеми звичайно заздалегідь визначається розробником системи, але агенти вільні у виборі способів досягнення своїх цілей.

Намагання ПА бути корисними для усіх навіть за рахунок власних інтересів є значним недоліком з точки зору їхньої автономності, тому що вони не враховують при виборі цілі витрачених ними зусиль на виконання потреб інших агентів. Хоча такий підхід надає багато корисних можливостей, він не дозволяє використовувати весь потенціал агентної парадигми. Зараз значна кількість розробників ПЗ *використовують більш конструктивний підхід до розробки МАС, у якому центральним об'єктом аналізу є окремі ПА, а не система в цілому.* Цей підхід більше відповідає специфіці відкритих і розподілених застосувань.

Таким чином, агентна парадигма привносить ряд принципово нових властивостей і можливостей в інформаційні технології і являє собою якісно новий рівень її розвитку, який забезпечується розподіленими обчисленнями в гетерогенному інформаційному середовищі Інтернету. Впровадження агентних технологій має підвищити ефективність інтелектуальної діяльності людини, позбавивши її від рутинних операцій. Одним з чинників інтересу до МАС став розвиток мережі Інтернету, в якій функціонують розподілені автономні програмні системи, що обробляють гетерогенні інформаційні ресурси.

Можна виділити такі основні напрямки застосування агентних технологій:

- пошук, фільтрація та аналіз інформації;
- ефективний доступ до Web-сервісів;
- ситуаційне керування;
- дистанційне навчання;
- електронний бізнес;
- моніторинг даних;
- телемедицина;
- керування виробництвом у режимі on-line;
- керування транспортними потоками.

3. Основи теорії нечітких множин

Найбільш вражаючою властивістю людського інтелекту є здатність приймати правильні рішення в умовах неповної і нечіткої інформації. Побудова моделей, які відтворюють мислення людини і використання їх у комп'ютерних системах на сьогодні є однією з найважливіших проблем науки.

При розробці інтелектуальних систем знання про конкретну предметну область, для якої створюється система, рідко бувають повними й абсолютно достовірними. Навіть кількісні дані, отримані шляхом досить точних експериментів, мають статистичні оцінки вірогідності, надійності, значимості і т. д. Інформація, якою заповнюються експертні системи, отримується у результаті опитування експертів, думки яких є суб'єктивними і можуть розходитися. Поряд із кількісними характеристиками в базах знань інтелектуальних систем повинні зберігатися якісні показники, евристичні правила, текстові знання і т. д. При обробці знань із застосуванням жорстких механізмів формальної логіки виникає протиріччя між нечіткими знаннями і чіткими методами логічного виведення. Розв'язати це протиріччя можна шляхом подолання нечіткості знань (коли це можливо) або використанням спеціальних методів подання й обробки нечітких знань.

Зміст терміна *нечіткість* багатозначний та включає такі основні компоненти: **недетермінованість виведень, багатозначність, ненадійність, неповнота, неточність.**

Недетермінованість виведень - це характерна риса більшості систем штучного інтелекту. Недетермінованість означає, що заздалегідь шлях вирішення конкретної задачі в просторі її станів визначити неможливо. Тому в більшості випадків методом проб і помилок вибирається деякий ланцюжок логічних висновків, що узгоджуються з наявними знаннями, а у випадку якщо він не приводить до успіху, організується перебір з поверненням для пошуку іншого ланцюжка і т. д. Такий підхід припускає визначення деякого первісного шляху. Для вирішення подібних задач запропоновано багато евристичних методів. Недетермінованість виведень варто враховувати при розробці ефективних способів подання і збереження знань, а також при побудові методів пошуку й обробки знань, що дозволяють одержати рішення задачі за найменше число кроків. Для побудови таких методів звичайно застосовуються евристичні *метазнання* (знання про знання).

Багатозначність інтерпретації - звичайне явище в задачах розпізнавання. При розумінні природної мови серйозними проблемами стають багатозначність змісту слів, їхня підпорядкованість, порядок слів у реченні і т. п. Проблеми розуміння змісту виникають у будь-якій системі, що взаємодіє з користувачем природною мовою. Розпізнавання графічних образів також пов'язано з вирішенням проблеми багатозначної інтерпретації. При комп'ютерній обробці знань багатозначність необхідно усувати шляхом вибору правильної інтерпретації, для чого розроблено спеціальні методи.

Ненадійність знань і виведень означає, що для оцінки вірогідності знань не можна застосувати двобальну шкалу (1 - абсолютно достовірні знання, 0 - недостовірні знання). Для більш тонкої оцінки вірогідності знань застосовується імовірнісний підхід, заснований на теоремі Байеса, і інші методи (наприклад, метод виведення з використанням коефіцієнтів упевненості). Широке застосування на практиці одержали нечіткі виведення, які будуються на базі нечіткої логіки, що веде своє походження від теорії нечітких множин.

Неповнота знань і немонотонна логіка. Абсолютно повних знань не буває, оскільки процес пізнання нескінченний. У зв'язку з цим стан бази знань повинний

змінюватися з часом. На відміну від простого додавання інформації, як у базах даних, при додаванні нових знань виникає небезпека одержання суперечливих висновків: тобто висновки, отримані з використанням нових знань, можуть спростовувати ті, що були отримані раніше. Ще гірше, якщо нові знання будуть знаходитися в протиріччі з старими, тоді механізм виведення може стати непрацездатним.

Більшість експертних систем першого покоління були засновані на моделі закритого світу, обумовленій застосуванням апарату формальної логіки для обробки знань.

Модель закритого світу припускає жорсткий добір знань, що включаються в базу, а саме: база знань заповнюється винятково вірними поняттями, а усе, що ненадійно або невиразно, свідомо вважається помилковим. Така модель має обмежені можливості подання знань і таїть у собі небезпеку одержання протиріч при додаванні нової інформації. Недоліки моделі закритого світу пов'язані з тим, що формальна логіка виходить з передумови, відповідно до якої набір визначених у системі аксіом (знань) є *повним* (теорія є повною, якщо кожний її факт можна довести, виходячи з аксіом цієї теорії). Для повного набору знань справедливості раніше отриманих виведень не порушується з додаванням нових фактів. Ця властивість логічних виведень називається *монотонністю*. На жаль, реальні знання, що закладаються в експертні системи, украй рідко бувають повними.

Неточність знань. Відомо, що кількісні дані (знання) можуть бути неточними, при цьому існують кількісні оцінки такої неточності (довірчий інтервал, рівень значимості, ступінь адекватності і т. д.). Лінгвістичні знання також можуть бути неточними. Для врахування неточності лінгвістичних знань використовується теорія нечітких множин. Фактично нечіткість може бути ключем до розуміння здатності людини справлятися з задачами, що занадто складні для вирішення на ЕОМ. Розвиток досліджень в області нечіткої математики призвів до появи нечіткої логіки і нечітких виведень, що виконуються з використанням знань, представлених нечіткими множинами, нечіткими відношеннями, нечіткими відповідностями і т. д.

Теорія нечітких множин (fuzzy sets theory) бере свій початок з 1965 року, коли професор Лотфі Заде (Lotfi Zadeh) з університету Берклі опублікував основну роботу «Fuzzy Sets» у журналі «Information and Control». Прикметник «fuzzy» (нечіткий, розмитий), введено в назву нової теорії з метою відокремлення від традиційної чіткої математики й аристотелевої логіки, що оперують з чіткими поняттями: «належить - не належить», «істина - хибність». Концепція нечіткої множини зародилася у Заде «як незадоволеність математичними методами класичної теорії систем, що змушувала домагатися штучної точності, недоречної в багатьох системах реального світу, особливо в так званих гуманістичних системах, що включають людей».

Системи з нечіткою логікою *доцільно застосовувати* для складних процесів, коли немає простої математичної моделі, а також якщо експертні знання про об'єкт або про процес можна сформулювати тільки в лінгвістичній формі.

Системи, що базуються на нечіткій логіці, *застосовувати недоцільно* якщо необхідний результат може бути отриманий яким-небудь іншим (стандартним) шляхом, або якщо для об'єкта або процесу вже знайдена адекватна і легко досліджувана математична модель.

Основні *недоліки систем з нечіткою логікою*: вихідний набір нечітких правил, що постулюються, формулюється експертом-людиною і може виявитися неповним або суперечливим; вид і параметри функцій приналежності, що описують вхідні і вихідні змінні системи, вибираються суб'єктивно і можуть виявитися такими, що цілком не відбивають реальну дійсність.

Нечіткі множини та змінні

Нехай E - універсальна множина, x - елемент E , а R - певна властивість. Звичайна (чітка) підмножина A універсальної множини E , елементи якої задовольняють властивості R , визначається як множина впорядкованої пари

$A = \{m_A(x)/x\}$, де $m_A(x)$ - характеристична функція, що приймає значення 1, якщо x задовольняє властивості R , і 0 - в іншому випадку.

Нечітка підмножина відрізняється від звичайної тим, що для елементів x з E немає однозначної відповіді "ні" відносно властивості R . У зв'язку з цим, нечітка підмножина A універсальної множини E визначається як множина впорядкованої пари $A = \{m_A(x)/x\}$, де $m_A(x)$ - **характеристична функція приналежності (або просто функція приналежності)**, що приймає значення в деякій впорядкованій множині M (наприклад, $M = [0,1]$).

Функція приналежності вказує ступінь (або рівень) приналежності елемента x до підмножини A . Множину M називають **множиною приналежностей**. Якщо $M = \{0,1\}$, тоді нечітка підмножина A може розглядатися як звичайна або чітка множина.

Нечітка змінна визначається як $\langle a, E, A \rangle$, де a - найменування змінної, $E = \{x\}$ - область визначення змінної, набір можливих значень x , $A = \{ \langle m_A(x)/x \rangle \}$ - нечітка множина, що описує обмеження на можливі значення змінної a (семантику). Нечітка змінна - це теж саме, що і нечітке число, тільки з додаванням імені, яким формалізується поняття, що описується цим числом.

Лінгвістична змінна визначається як $\langle B, T, X, G, M \rangle$, де B - найменування змінної, T - множина її значень (базова терм-множина), що складається з найменувань нечітких змінних, областю визначення кожної з яких є множина X ; G - синтаксична процедура (граматика), що дозволяє оперувати елементами терм-множини T , зокрема - генерувати нові осмислені терми; $T = T \cup G(T)$ задає розширену терм-множину (\cup - знак об'єднання); M - семантична процедура, що дозволяє приписати кожному новому значенню лінгвістичної змінної нечітку семантику, шляхом формування нової нечіткої множини.

Лінгвістична змінна - це множина нечітких змінних, вона використовується для того, щоб дати словесний опис деякому нечіткому числу, отриманому в результаті деяких операцій.

Терм-множина - це множина всіх можливих значень лінгвістичної змінної.

Терм - будь-який елемент терм-множини. У теорії нечітких множин терм формалізується нечіткою множиною за допомогою функції приналежності.

Нечіткий терм - це нечітка множина, яка має властивість, якій відповідає певне поняття.

Наочним прикладом нечіткої логіки можна навести відповіді людей на питання: «Чи холодно вам зараз?». В більшості випадків люди розуміють, що мова не йде про абсолютну температуру за шкалою Цельсія, а про особисте сприйняття температури. Для

багатьох людей $+15^{\circ}\text{C}$ буде цілком теплою, для інших така температура буде трактуватися як прохолодна.

На відміну від людей, машини не здатні проводити таку тонку градацію. Якщо стандартом визначення холоду буде «температура нижче $+15^{\circ}\text{C}$ », то $+14,99^{\circ}\text{C}$ буде розцінюватися як холод, а $+15^{\circ}\text{C}$ - не буде.

Базові концепції нечіткої логіки є доволі простими. На рис. 1. представлено графік, що допомагає зрозуміти те, як людина сприймає температуру. Температуру в $+10^{\circ}\text{C}$ людина сприймає як холод, а температуру в $+30^{\circ}\text{C}$ - як спеку. Температура в $+15^{\circ}\text{C}$ одним здається низькою, іншим - достатньо комфортною. Назвемо цю групу визначень функцією приналежності до множин, які описують суб'єктивне сприйняття температури людиною.

Аналогічно можна створити додаткові множини, що описують сприйняття температури людиною. Наприклад, можна додати такі множини, як «дуже холодно» і «дуже жарко». Можна описати подібні функції для інших концепцій, наприклад, для станів «відкрито» і «закрито», температури в охолоджувачі або температури в теплиці.

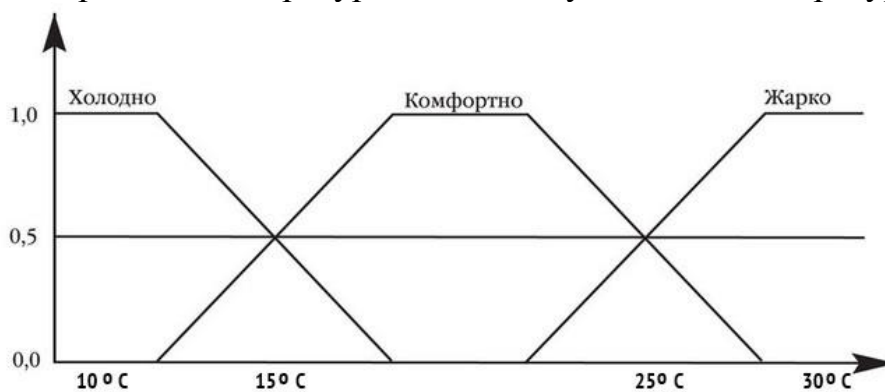


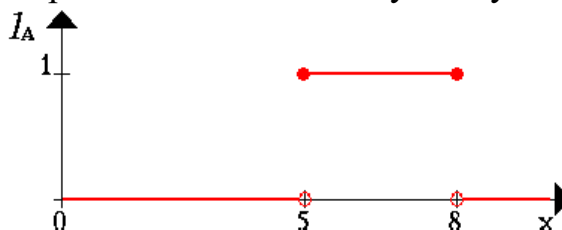
Рис.1. Нечітке визначення температури

Тобто, нечіткі системи можна використовувати як **універсальний апроксиматор (усереднювач)** дуже широкого класу лінійних і нелінійних систем. Це не лише робить більш надійними стратегії контролю в нелінійних випадках, але і дозволяє використовувати оцінки фахівців-експертів для побудови схем комп'ютерної логіки.

Розглянемо множину X всіх чисел від 0 до 10. Визначимо підмножину A множини X всіх дійсних чисел від 5 до 8.

$$A = [5,8]$$

Покажемо функцію приналежності множини A , ця функція ставить у відповідність число 1 чи 0 кожному елементу в X , у залежності від того, належить даний елемент підмножині A чи ні. Результат представлений на наступному малюнку:



Можна інтерпретувати елементи, яким поставлена у відповідність 1, як елементи, що знаходяться в множині A , а елементи, яким поставлено у відповідність 0, як елементи, що не знаходяться в множині A .

Ця концепція використовується в багатьох областях застосувань. Але можна легко знайти ситуації, в яких даній концепції буде бракувати гнучкості.

Наприклад, опишемо множину молодих людей. Формально можна записати так $B = \{\text{множина молодих людей}\}$

Оскільки, вік починається з 0, то нижня межа цієї множини повинна бути нулем. Верхню межу визначити небагато складніше. Спочатку встановимо верхню межу, наприклад 20 років. Таким чином, маємо B як чітко обмежений інтервал, буквально: $B=[0,20]$. Виникає питання: чому людина в двадцятирічній ювілей - молода, а наступного дня вже не молода? Очевидно, це структурна проблема, і якщо пересунути верхню межу в іншу точку, то можна задати таке ж питання.

Більш природний шлях отримання множини B складається в послабленні строгого поділу на молодих і не молодих.

Зробимо це, виносячи не лише чіткі судження:

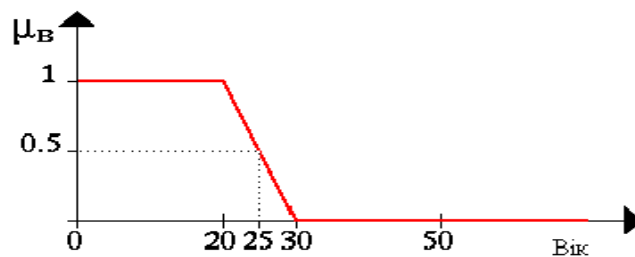
- Так, він належить до множини молодих людей
- Ні, він не належить до множини молодих людей, але і більш гнучкі формулювання
- Так, він належить до досить молодих людей
- Ні, він не дуже молодий.

Розглянемо як за допомогою нечіткої множини визначити такий вираз, як він ще молодий.

В першому прикладі ми кодували всі елементи множини за допомогою 0 чи 1. Простим способом узагальнити дану концепцію є введення значення між 0 і 1. Реально можна навіть допустити нескінченне число значень між 0 і 1, в одиничному інтервалі $I = [0, 1]$.

Інтерпретація чисел при співвідношенні всіх елементів множини стає тепер більш складною. Звичайно, знову число 1 ставиться у відповідність до того елемента, що належить множині B , а 0 означає, що елемент точно не належить множині B . Всі інші значення визначають ступінь приналежності до множини B .

Для наочності приведемо характеристичну функцію множини молодих людей, як і в першому прикладі.



Нехай $E = \{x_1, x_2, x_3, x_4, x_5\}$, $M = [0,1]$; A - нечітка множина, для якої $m_A(x_1)=0,3$; $m_A(x_2)=0$; $m_A(x_3)=1$; $m_A(x_4)=0,5$; $m_A(x_5)=0,9$

Тоді A можна представити у виді:

$A = \{0,3/x_1; 0/x_2; 1/x_3; 0,5/x_4; 0,9/x_5\}$ або

$A = 0,3/x_1 + 0/x_2 + 1/x_3 + 0,5/x_4 + 0,9/x_5$,

(знак "+" є операцією не додавання, а об'єднання) або

	x_1	x_2	x_3	x_4	x_5
$A =$	0,3	0	1	0,5	0,9

4. Функції приналежності та операції над нечіткими множинами

Функції приналежності нерозривно пов'язані із нечіткими множинами. Тип функції приналежності в значному ступені визначає властивості нечіткої системи.

Задавання функцій приналежності можна здійснювати у вигляді списку з явним перерахуванням усіх елементів та відповідних ним значень функції приналежності (наприклад, використовуючи відносні частоти за даними експерименту як значення приналежності), або аналітично у вигляді формул (наприклад, використовуючи типові форми кривих для завдання функцій приналежності з уточненням їхніх параметрів відповідно до даних експерименту).

Існують прямі та непрямі *методи побудови функцій приналежності*.

При використанні *прямих методів* експерт просто задає для кожного $x \in E$ значення $\mu(x)$. Як правило, прямі методи побудови функції приналежності використовуються для вимірних понять, таких як швидкість, час, відстань, тиск, температура і т. д., або тоді, коли виділяються полярні значення.

У багатьох задачах при характеристиці об'єкта можна виділити набір ознак і для кожної з них визначити полярні значення, що відповідають значенням функції приналежності 0 або 1. Для конкретного об'єкта експерт, виходячи з приведеної шкали, задає $\mu_A(x) \in [0, 1]$, формуючи векторну функцію приналежності $\{\mu_A(x_1), \mu_A(x_2), \dots, \mu_A(x_n)\}$.

Різновидом прямих методів побудови функцій приналежності є *прямі групові методи*, коли, наприклад, групі експертів пред'являють конкретний об'єкт, і кожен повинний дати одну з двох відповідей: належить чи не належить цей об'єкт до заданої множини. Тоді число позитивних відповідей, поділене на загальне число експертів, дає значення функції приналежності об'єкта до даної нечіткої множини.

Непрямі методи визначення значень функції приналежності використовуються у випадках, коли немає вимірних елементарних властивостей, через які визначається нечітка множина. Як правило, *це методи попарних порівнянь*. Якщо значення функцій приналежності відомі, наприклад, $\mu_A(x_i) = w_i$, $i = 1, 2, \dots, n$, то попарні порівняння можна подати матрицею відношень $A = \{a_{ij}\}$, де $a_{ij} = w_i/w_j$ (операція розподілу).

На практиці експерт сам формує матрицю A , при цьому передбачається, що діагональні елементи дорівнюють 1, а для елементів, симетричних щодо головної діагоналі, $a_{ij} = 1/a_{ji}$ тобто якщо один елемент оцінюється як в a разів більш значущий ніж інший, то цей останній повинний бути в $1/a$ разів більш значущим, ніж перший. У загальному випадку задача зводиться до пошуку вектора w , що задовольняє рівнянню виду: $Aw = \lambda_{\max} w$, де λ_{\max} - найбільше власне значення матриці A . Оскільки матриця A позитивна за побудовою, розв'язок даної задачі існує і є позитивним.

Обмеженням методів попарного порівняння є використання суб'єктивної інформації і деяких допущень при перетворенні її в ступені приналежності нечітких множин.

Оптимізаційні методи побудови функцій приналежності засновані на параметричній ідентифікації нечітких моделей за експериментальними даними «входи - вихід», при якій оптимізують параметри функцій приналежності з метою мінімізації відхилення між експериментальними даними і результатами нечіткого моделювання. Використання оптимізаційного підходу знімає суб'єктивізм побудови функцій приналежності, однак замість цього вимагає навчаючої вибірки та нечіткої моделі «входи - вихід». Недоліком даних методів є те, що функції приналежності однакових за змістом нечітких множин виходять різними в

результаті ідентифікації різних залежностей «входи -вихід». Таким чином, функція приналежності стає сильно чутливою до навчаючої вибірки та структури нечіткої моделі.

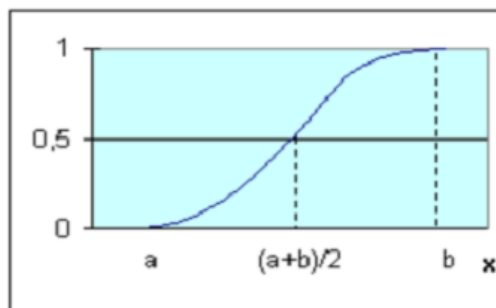
Існує і більш складний - Метод побудови функцій приналежності шляхом кластеризації експериментальних даних. (У лекції ми його розглядати не будемо)

Візуалізувати функції приналежності нечітких множин можна шляхом побудови графіку залежності значення функції приналежності μ від значення елемента нечіткої множини x .

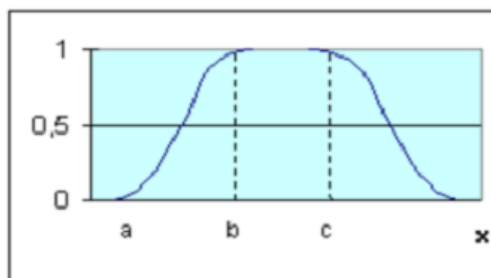
Виділяють такі основні типи функцій приналежності: кусочно-лінійні функції, Z-подібні та S-подібні функції, П--подібні функції.

Приклади:

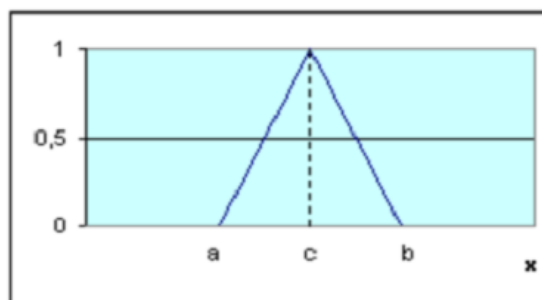
$$\mu_1(x, a, b) = \begin{cases} 0, & \text{если } x \leq a; \\ \frac{2(x-a)^2}{(b-a)^2}, & \text{если } a < x \leq \frac{a+b}{2}; \\ 1 - \frac{2(x-a)^2}{(b-a)^2}, & \text{если } \frac{a+b}{2} < x < b; \\ 1, & \text{если } x \geq b. \end{cases}$$



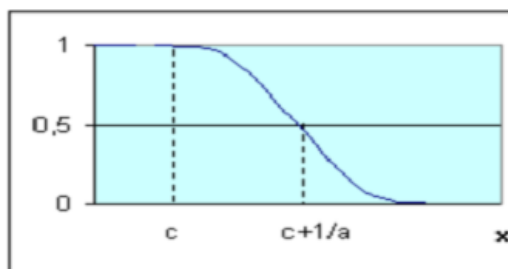
$$\mu_2(x, a, b, c) = \begin{cases} \mu_1(x, a, b), & \text{если } x < b; \\ 1, & \text{если } b \leq x \leq c; \\ 1 - \mu_1(x, c, c+b-a), & \text{если } x > c. \end{cases}$$



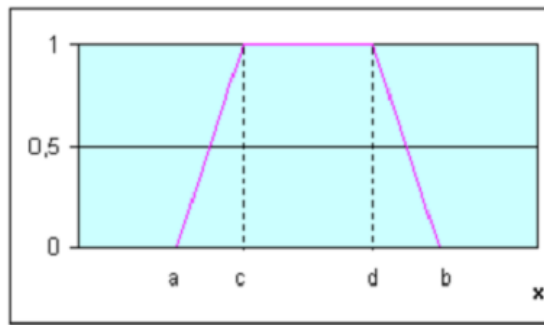
$$\mu_3(x, a, b, c) = \begin{cases} 0, & \text{если } x \leq a; \\ \frac{x-a}{c-a}, & \text{если } a < x \leq c; \\ \frac{b-x}{b-c}, & \text{если } c < x < b; \\ 0, & \text{если } x \geq b. \end{cases}$$



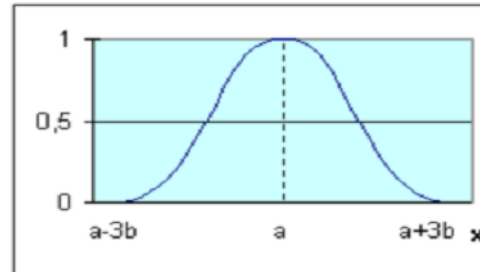
$$\mu_4(x, a, b, c) = \begin{cases} 1, & \text{если } x \leq c; \\ \{1 + [a(x-c)]^b\}^{-1}, & \text{если } x > c. \end{cases}$$



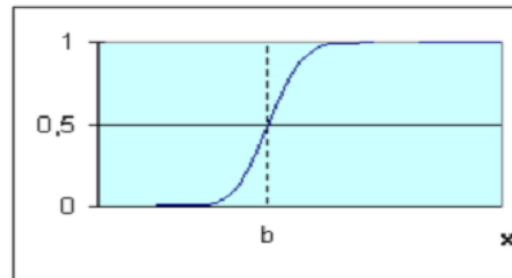
$$\mu_5(x, a, b, c, d) = \begin{cases} 0, & \text{если } x \leq a; \\ \frac{x-a}{c-a}, & \text{если } a < x < c; \\ 1, & \text{если } c \leq x \leq d; \\ \frac{b-x}{b-d}, & \text{если } d < x < b; \\ 0, & \text{если } x \geq b. \end{cases}$$



$$\mu_6(x, a, b) = \exp\left[-\frac{(x-a)^2}{2b^2}\right]$$



$$\mu_7(x, a, b) = \{1 + \exp[-a(x-b)]\}^{-1}$$



Операції над нечіткими множинами

Щоб застосувати алгебру для роботи з нечіткими значеннями, потрібно визначити оператори, що будуть використовуватися. Зазвичай, в булевій логіці використовується лише обмежений набір операторів, за допомогою яких і проводиться виконання інших операцій: AND (оператор «І»), OR (оператор «АБО»), NOT (оператор «НЕ»).

AND	0	1
0	0	1
1	1	1

OR	0	1
0	0	0
1	0	1

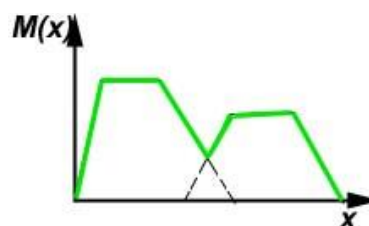
A	0	1
NOT A	1	0

Можна дати багато визначень для операторів, три базових з яких наведено в таблиці.

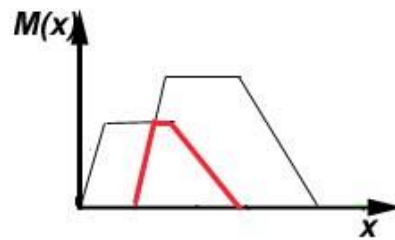
У булевій логіці значення FALSE («ХИБНІСТЬ») еквівалентно значенню «0», а значення TRUE («ІСТИНА») еквівалентно значенню «1». Аналогічним чином в нечіткій логіці ступінь істинності може змінюватися в діапазоні від 0 до 1, тому значення «Холод» вірно в ступені 0,1, а операція NOT («Холод») дасть значення 0,9.

Операції над нечіткими множинами

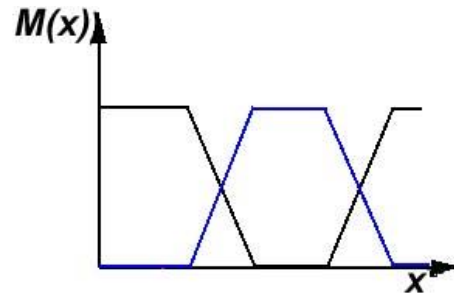
Об'єднання



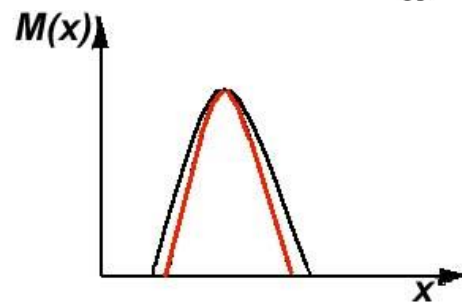
Перетин



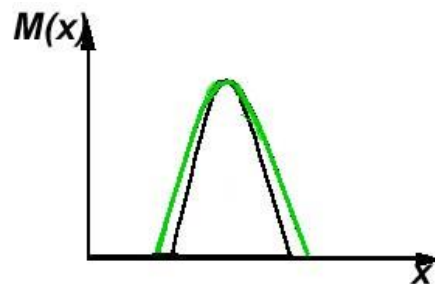
Доповнення



Концентрація



Розмивання



5. Практичне застосування нечіткої логіки

Вдалим застосуванням теорії нечітких множин є контролери нечіткої логіки. Їх функціонування дещо відрізняється від роботи звичайних контролерів; для опису системи замість диференціальних рівнянь використовуються знання експертів. Ці знання можуть бути виражені за допомогою лінгвістичних змінних, які описані нечіткими множинами.

Фазифікація - зіставлення множини значень x з її функцією приналежності $M(x)$, тобто переведення значень x в нечіткий формат (приклад з терміном молодий).

Дефазифікація - процес, зворотний до фазифікації.

Всі системи з нечіткою логікою функціонують за одним принципом: показання вимірювальних приладів фазифікуються (переводяться в нечіткий формат), обробляються, дефазифікуються і у вигляді звичних сигналів подаються на виконавчі пристрої.

Ступінь приналежності - це не ймовірність, так як невідома функція розподілу, немає повторюваності експериментів. Так, якщо взяти з прикладу прогнозу погоди дві взаємовиключні події: буде дощ і не буде і присвоїти їм деякі ранги, то сума цих рангів необов'язково буде дорівнювати 1, але якщо рівність все-таки є, то нечітка множина вважається нормованою. Значення функції приналежності $M(x)$ можуть бути взяті тільки з апіорних знань, інтуїції (досвіду), опитування експертів.

Загальна структура нечіткого мікроконтролера

Загальна структура мікроконтролера, що використовує нечітку логіку, показана на рис. Вона містить у своєму складі наступні складові:

Блок фазифікації.

База знань.

Блок рішень.

Блок дефазифікації.

Блок фазифікації перетворює чіткі величини, які виміряні на виході об'єкта керування у нечіткі величини, що описані лінгвістичними змінними в базі знань.

Блок рішень використовує нечіткі умовні (if - then) правила, що закладено в базу знань, для перетворення нечітких вхідних даних в керуючі впливи, які мають також нечіткий характер.

Блок дефазифікації перетворює нечіткі дані з виходу блоку рішень в чіткі величини, які використовуються для керування об'єктом.

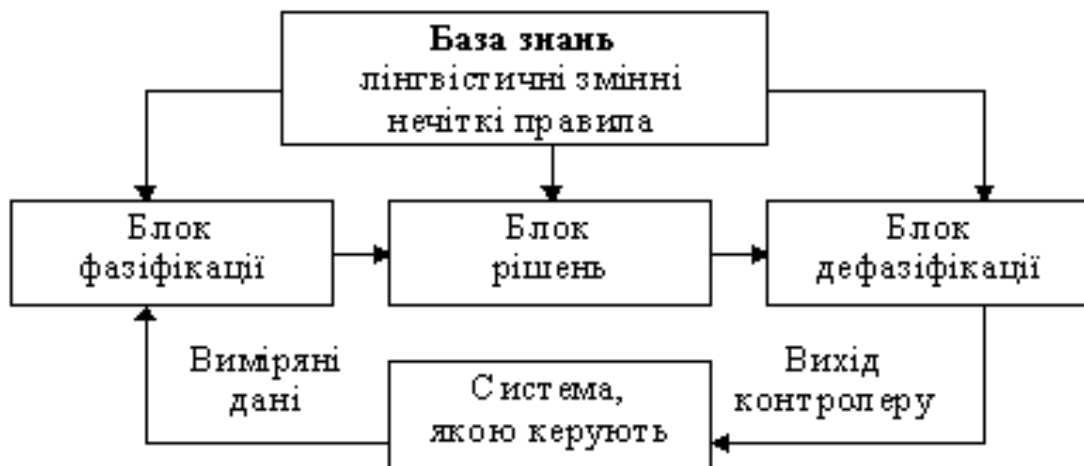


Рис. 1. Загальна структура нечіткого мікроконтролера

Розглянемо випадок керування мобільним роботом, задачею якого є об'їзд перешкоди. Введемо дві лінгвістичні змінні: ДИСТАНЦІЯ (відстань від робота до перешкоди) і НАПРЯМОК (кут між подовжньою віссю робота та напрямком до перешкоди).

Розглянемо лінгвістичну змінну ДИСТАНЦІЯ. Значеннями її можна визначити терми ДАЛЕКО, СЕРЕДНЬО, БЛИЗЬКО і ДУЖЕ БЛИЗЬКО. Для фізичної реалізації лінгвістичної змінної необхідно визначити точні фізичні значення термів цієї змінної. Нехай змінна ДИСТАНЦІЯ може приймати будь-які значення з діапазону від нуля до нескінченності. Відповідно до теорії нечітких множин, в такому випадку кожному значенню відстані із зазначеного діапазону може бути поставлене у відповідність деяке число від нуля до одиниці, що визначає ступінь приналежності даної фізичної відстані (припустимо 40 см) до того чи іншого терму лінгвістичної змінної ДИСТАНЦІЯ.

Ступінь приналежності визначаємо функцією приналежності $M(d)$, де d -відстань до перешкоди. В нашому випадку відстань 40 см. Можна задати ступінь приналежності до терму ДУЖЕ БЛИЗЬКО, що дорівнює 0,7, а до терму БЛИЗЬКО - 0,3 (рис. 2.).

Конкретне визначення ступеня приналежності визначається експертами.

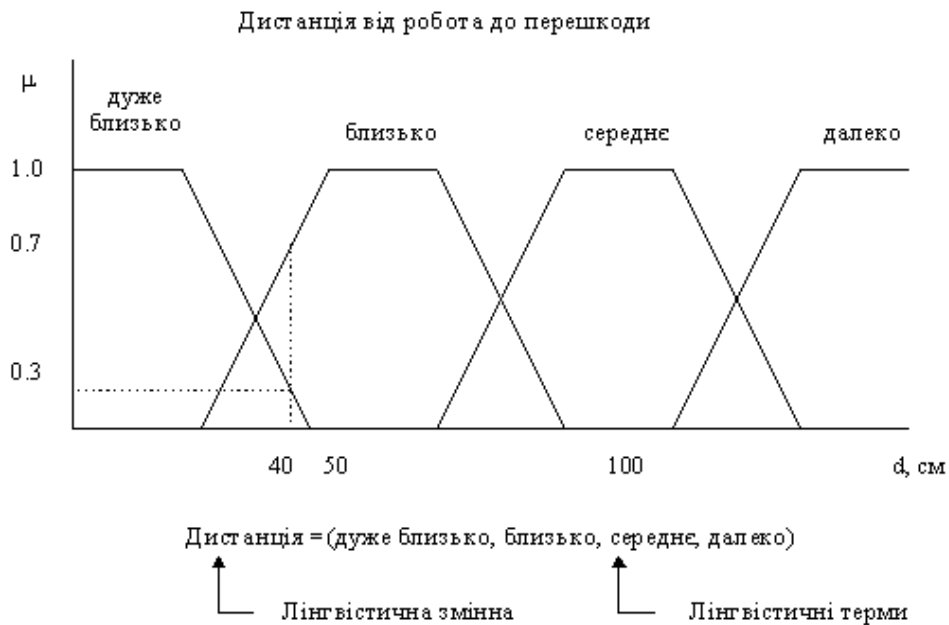


Рис. 2. Лінгвістична змінна і функція приналежності

Змінній НАПРЯМОК, яка може приймати значення в діапазоні від 0 до 360 градусів, задамо терми ВЛІВО, ПРЯМО і ВПРАВО.

Тепер необхідно задати вихідні змінні. У даному прикладі достатньо однієї, яку назовемо КУТ ПОВОРОТУ. Вона може містити терми: РІЗКО ВЛІВО, ВЛІВО, ПРЯМО, ВПРАВО, РІЗКО ВПРАВО. Зв'язок між входом та виходом запам'ятовується в таблиці нечітких правил.

Таблиця нечітких правил
дiстанцiя

		дуже близько	близько	середнє	далеко
напрямок	правий	різко вліво	різко вліво	вліво	прямо
	прямий	різко вліво	вліво	вліво	прямо
	лівий	різко вправо	різко вправо	вправо	прямо

Кожний запис в даній таблиці відповідає своєму нечіткому правилу, наприклад: Якщо дистанція до перешкоди - «близько» і напрямок «правий», тоді кут повороту «різко вліво».

Таким чином, мобільний робот з нечіткою логікою буде працювати за наступним принципом: дані з сенсорів про відстань до перешкоди та напрямок до неї будуть фазифіковані, оброблені згідно табличних правил, дефазифіковані і отримані дані у вигляді керуючих сигналів надходять на приводи робота.

Переваги нечітких систем

- Можливість оперувати входними даними, заданими нечітко: наприклад, дані, які неперервно змінюються в часі (динамічні задачі), значення, що неможливо задати однозначно (результати статистичних опитувань, рекламні компанії);

- Можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями "більшість", "можливе", "переважно" тощо.;
- Можливість проведення якісних оцінок як вхідних даних, так і виведених результатів: значення даних, їх ступень достовірності (не плутати з імовірністю!) та її розподілом;
- Можливість проведення швидкого моделювання складних динамічних систем та їх порівняльний аналіз із заданим ступенем точності: оперуючи принципами поведінки системи, описаними fuzzy-методами:
 - можна швидко з'ясувати точні значення змінних і скласти правила, що їх описують,
 - можна оцінити різні варіанти вихідних значень.

Сьогодні елементи нечіткої логіки можна знайти в багатьох промислових виробках - від систем керування електропоїздами і бойовими вертольотами до побутової техніки. Без застосування нечіткої логіки немислимі сучасні ситуаційні центри керівників західних країн, в яких приймаються ключові політичні рішення і моделюються всілякі кризові ситуації.

Активними споживачами нечіткої логіки є банкіри і фінансисти, а також фахівці в області політичного й економічного аналізу, задачі яких вимагають щоденного прийняття правильних рішень у складних умовах непередбаченого ринку. Вони використовують нечіткі системи для створення моделей різних економічних, політичних, біржових ситуацій.

Пристрої для автоматичної підтримки швидкості руху автомобіля і збільшення ефективності / стабільності роботи автомобільних двигунів (компанії Nissan, Subaru).

Системи розпізнавання рукописного тексту в PDA (компанія Sony).

Поліпшення систем безпеки для атомних реакторів (компанії Hitachi, Bernard, Nuclear Fuel Div.).

Управління роботами (компанії Toshiba, Fuji Electric, Omron).

Промислові системи управління (компанії Apronix, Omron, Meiden, Sha, Micom, Nisshin-Denki, Mitsubishi, Oku-Electronics та ін.).