УДК 681.3, 621.3

Г.В. Кицун
Національний університет "Львівська політехніка",
кафедра електронних обчислюваних машин

АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОГО АГЕНТА

Запропоновано вирішення проблеми обміну знаннями між окремими модулями архітектури інтелектуального агента. Запропоновано архітектуру та процес узгодження рівнів інтелектуального агента. Наведено класифікацію та порівняння існуючих архітектур агента. Запропонований механізм взаємодії блоку планування агента.

In this article solution of knowledge's exchange problem between the separate modules of intellectual agent architecture is offered. Architecture and concordance process of intellectual agent levels is offered. Classification and comparison of existent agent architectures is resulted. The mechanism of agent planning block co-operation is offered.

Вступ

Останнім часом все більше уваги приділяють розподіленим інтелектуальним системам, які обмінюються знаннями одна з однією, використовують спільні знання, реалізують можливості повторного використання знань. Сучасні робототехнічні системи складаються з множини досить самостійних модулів. Такий модуль, який часто є автономною частиною системи, можна розглядати як агент, що працює в межах цієї системи і який обмінюється знаннями з іншими її модулями за допомогою повідомлень [11].

Істотний ріст складності комп'ютерних систем висуває вимоги до вищої надійності, гнучкості, маштабованості, адаптивності й здатності системи до інтеграції. Агенти й багатоагентні системи породжують новий підхід до розвитку складних систем програмного забезпечення загалом й апаратних систем зокрема. Через такі принципи, як децентралізація, автономія, орієнтація на мету, реактивність і проактивність індивідуальних агентів та взаємодія між агентами, агентно-орієнтований підхід спроможний виконати вищезгадані вимоги.

Методи координації й співпраці повинні відповідати поняттям агентно-орієнтованої методології й здійснювати спільну діяльність агентів. Ці методи повинні також брати до уваги особливості системи, що розробляється, й середовища, до якого застосовують цю систему. За дотримання цих умов координація між агентами може значно збільшити ефективність колективної дії агентів.

Постановка задачі

Задачею є дослідження в галузі планування, процесу роботи децентралізованої інтелектуальної системи. Проаналізувати та класифікувати вже існуючі архітектури інтелектуального агента, відомі з літератури і дослідити, як ці механізми можна застосувати до системи колективної поведінки. За результатами аналізу запропонувати концепцію механізмів взаємодії блоку планування інтелектуального агента.

Огляд літературних джерел

Під інтелектуальними агентами в інформатиці і ШІ розуміють будь-які фізичні або віртуальні одиниці [10]:

- здатні діяти на об'єкти в деякому середовищі, на інших агентів, а також на самих себе (дія);
- здатні спілкуватися з іншими агентами (спілкування);
- які випливають з деяких потреб і здатні до цілеутворення (цільова основа); які володіють набором інтенціональних характеристик (переконання, бажання, наміри та ін.);

- які несуть певні обов'язки і що надають якісь послуги (наявність зобов'язань);
- які володіють власними ресурсами, що забезпечують їхню автономію (автономія);
- здатні до сприйняття середовища (сприйняття з обмеженим дозволом);
- здатні будувати часткове представлення цього середовища на основі її сприйняття, тобто перцептивних навиків і умінь (локальне представлення середовища);
- здатні до навчання, еволюції й адаптації (еволюційний і адаптаційний потенціал);
- здатні до самоорганізації і самовідтворення (самозбереження).

Уточнимо деякі поняття, які використовуватимуться надалі.

На рис. 1 показано структуру взаємодії середовища, об'єкта і системи управління інтелектуального агента [1].

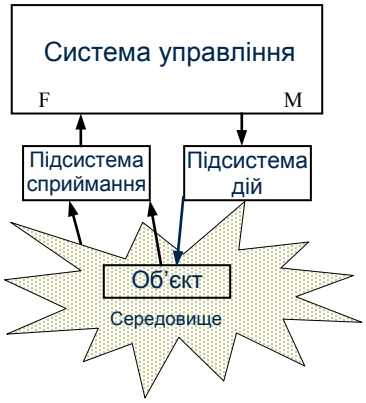

Рис. 1. Взаємодія агента і середовища

Відмінність середовища від об'єкта полягає в тому, що на середовище система управління агента безпосередньо не впливає, хоча опосередковано вона може діяти на середовище через об'єкт. Система управління агента одержує інформацію від середовища та об'єкта через підсистему сприймання. Рішення, шо формуються системою управління інтелектуального агента, надходять до об'єкта через підсистему дії агента.

Практично будь-яка система управління інтелектуального агента повинна будуватися на основі певних знань про об'єкт управління. Вони відображають фундаментальні властивості об'єкта, не залежні від поточної ситуації. Безліч цих знань надалі називатиметься моделлю знань агента про об'єкт. Аналогічну інформацію система управління агента може мати і про середовище, що дає можливість говорити про модель знань середовища. Ця модель знань спільно з моделлю знань про об'єкт називатиметься просто моделлю знань агента. У системі управління агента завжди присутній деякий механізм породження рішень у вигляді сукупності певних процедур. Цей механізм надалі називатимемо підсистемою планування агента.

Архітектура інтелектуального агента

Розглянемо стисло архітектуру взаємодії агентів. Введемо поняття архітектури: архітектура – це структура і поведінка агента з погляду сусіднього агента. Можна виділити два основні варіанти архітектури. У одному з них агенти не утворюють ієрархії і розв'язують загальну задачу повністю в розподіленому варіанті. В іншому варіанті координація розподіленого функціонування агентів тією чи іншою мірою підтримується спеціально виділеним агентом, який при цьому належить до метарівня відносно решти агентів.

Розрізняють два основні класи архітектури [2]:

– архітектура, яка ґрунтується на принципах і методах штучного інтелекту, тобто систем, заснованих на знаннях ("deliberative agent architecture", "архітектура розумного агента");

– архітектура, яка базується на поведінці (reactive architecture), або "реактивна архітектура" (заснована на реакції системи на події зовнішнього світу).

З іншого боку, архітектуру агентів класифікують залежно від виду структури, накладеної на функціональні компоненти агента і прийнятих методів організації взаємодії його компонент в процесі роботи. Як правило, архітектуру агента представляють у вигляді декількох рівнів. Відповідно до [8], серед багаторівневої архітектури розрізняють горизонтальну і вертикальну організації взаємодії рівнів.

З класичної точки зору [7] архітектура на основі знань – це така архітектура, яка містить символьну модель світу, подану в явній формі і в якій рішення про дії, які повинні виконуватися агентом, ухвалюють на основі міркувань логічного типу. Такий агент можна розглядати як спеціальний випадок системи, заснованої на знаннях.

Архітектуру на основі планування (“плануючий агент”) розглядають як альтернативну попередній.

За цим підходом планування розглядали як “конструювання послідовності дій, яка, будучи виконаною, приводила б в результаті до досягнення бажаної мети” [7]. Простим прикладом архітектури такого роду є архітектура, в якій реакція агента на зовнішні події генерується кінцевим автоматом. Як інший приклад системи з цією архітектурою можна розглядати і широковідому систему STRIPS [6]. У цій системі, як відомо, використовували суто логічний підхід спільно з передумовами і післяумовами, що асоціюються з кожною з дій. Відповідно до прийнятої стратегії STRIPS, маючи опис світу і бажану мету, намагаються знайти послідовність дій, яка у результаті приведе до досягнення мети із задоволенням післяумов. Як відомо, система виявилася вкрай неефективною. Пізніше було розроблено й інші подібні підходи, проте вони не могли працювати із завданнями, в яких були темпоральні обмеження і обмеження реального часу, вельми істотні для застосувань інтелектуальних агентів.

Тільки найпростіші застосування агентів можуть бути реалізовані за допомогою однорівневої схеми. Як правило, функціональні модулі агента структуруються в декілька рівнів, проте за різним принципом. Як правило, рівні представляють різні функціональності, такі, як сприйняття зовнішніх подій і прості реакції на них; поведінка, керована цілями; координація поведінки з іншими агентами; оновлення внутрішнього стану агента, тобто переконань про зовнішній світ; прогнозування станів зовнішнього світу; визначення своїх дій на черговому кроці та ін. Найчастіше в архітектурі агента присутні рівні, що відповідають за [9]: сприйняття і виконання дій, реактивну поведінку, локальне планування, колективну поведінку, моделювання, формування намірів, навчання агента.

Існує два основні класи багаторівневої архітектури залежно від того, як організовується взаємодія рівнів [9]:

– горизонтально організована архітектура;
– вертикально організована архітектура.

У першій з них – в горизонтально організованій архітектурі – всі рівні агента мають доступ до рівня сприйняття і дій (загалом – всі рівні можуть спілкуватися між собою в стилі “бродкастингу”). Варіант такої архітектури наведено на рис. 2, *а*. У вертикально організованій архітектурі тільки один з рівнів має доступ до рівня сприйняття і дій, а кожний з решти рівнів спілкується тільки з парою безпосередньо суміжних з ним рівнів. Приклади такої архітектури наведено на рис. 2, *б*, *в*.

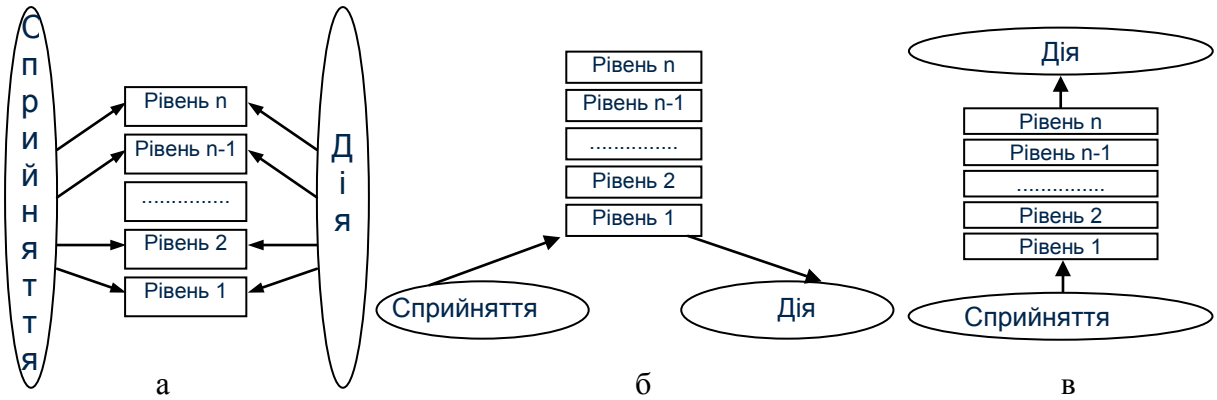

Рис. 2. Архітектура агента

Прикладами горизонтально організованої архітектури є архітектура Touring Machine [5] і Will-architecture [4] (D.Moffat and N.H.Frijda). Вдалим прикладом вертикально організованої архітектури є InteRRaP-архітектура [3].

Отримані результати

Порівняння архітектури інтелектуального агента

Основні проблеми реалізації горизонтально організованої архітектури обумовлені складністю організації узгодженої роботи всіх рівнів. В архітектурі Touring Machine ця проблема вирішується за допомогою спеціального алгоритму, який пригнічує входи деяких рівнів, якщо відповідна інформація не має до них відношення і цензурує виходи. Це виконується за допомогою спеціального набору правил. У другій з горизонтально організованих архітектур – архітектурі Will – завдання управління узгодженою роботою рівнів виконується за допомогою введення спеціальних функцій сумісності вхідних подій з "інтересами" (concerns) рівнів. Тут робиться спроба ввести деяку самоорганізацію, проте з наявних робіт не цілком зрозуміло, як це можна реалізувати в різних застосуваннях.

У вертикально організованій архітектурі проблема управління взаємодією рівнів не є такою складною, оскільки вихідна інформація кожного з рівнів завжди має адресата. У відомій вертикально організованій архітектурі функціональні модулі розподіляють за рівнями за одним з двох принципів. За одним з них різні рівні відповідають різному рівню абстракції, в основному, одного і того ж набору функціональностей (такий принцип використано у вже згадуваній InteRRap-архітектурі). За іншим принципом кожен рівень відповідає деякій функціональності або їхньому набору. За таким принципом побудовано MECCA-архітектуру [3], в якій цикл функціонування агента складається з чотирьох фаз: активація мети, планування, конкретизація плану в набір дій і виконання. Відповідно до цих фаз архітектура агента складається з чотирьох рівнів.

Недоліком вертикально організованої архітектури вважають таку її властивість, як переобтяжений рівень виконання (дій). Наприклад, в IntRRaP- архітектурі нижній рівень повинен реагувати на непередбачені події, відстежувати виконання команд, одержаних з рівня локального планування, стежити за виконанням обмежень, накладених контекстом локального планування (часових, ресурсних) і, нарешті, він повинен функціонувати відповідно до додаткових колективних обов'язків (зобов'язань), які покладені на агента іншими агентами багатоагентної системи.

Гібридна архітектура інтелектуального агента

Основна ідея розробленої архітектури полягає в тому, щоб представити агента як безліч рівнів, які зв'язані через керівну структуру і використовують загальну базу знань. Цю архітектуру наведено на рис. 3. Вона складається з кількох основних частин: інтерфейсу із зовнішнім світом; компоненти, заснованої на поведінці; плануючої компоненти; компоненти, відповідальної за кооперацію з іншими агентами і бази знань агента.

Інтерфейс із зовнішнім світом містить можливості агента із сприйняття подій зовнішнього світу, дії на нього і засоби комунікації.

Компонента, відповідальна за реактивну поведінку, використовує базові можливості агента щодо реактивної поведінки, а також частково використовує знання агента процедурного характеру. Вона ґрунтується на понятті "фрагмента поведінки" як деякої заготовки реакції агента на деякі стандартні ситуації. Це дає змогу агенту в стандартних ситуаціях не звертатися до планування на основі знань і реалізовувати значну частину своєї поведінки з хорошою ефективністю. З бази знань їй доступні тільки знання нижнього рівня абстракції, де міститься інформація про фрагменти поведінки.

Компонента, відповідальна за планування, містить механізм планування, що дає змогу будувати локальні плани агента, тобто плани, не пов'язані з колективною поведінкою. План подають у вигляді графу, вузлами якого можуть бути або конкретні набори дій аж до елементарних кроків поведінки, або нові підплани, що надалі конкретизують. Отже, планувальна компонента активізує поведінку, що керується цілями. Вона ж бере участь і в плануванні, пов'язаному з колективною поведінкою агентів. Ця компонента може використовувати знання двох нижніх рівнів абстракції.

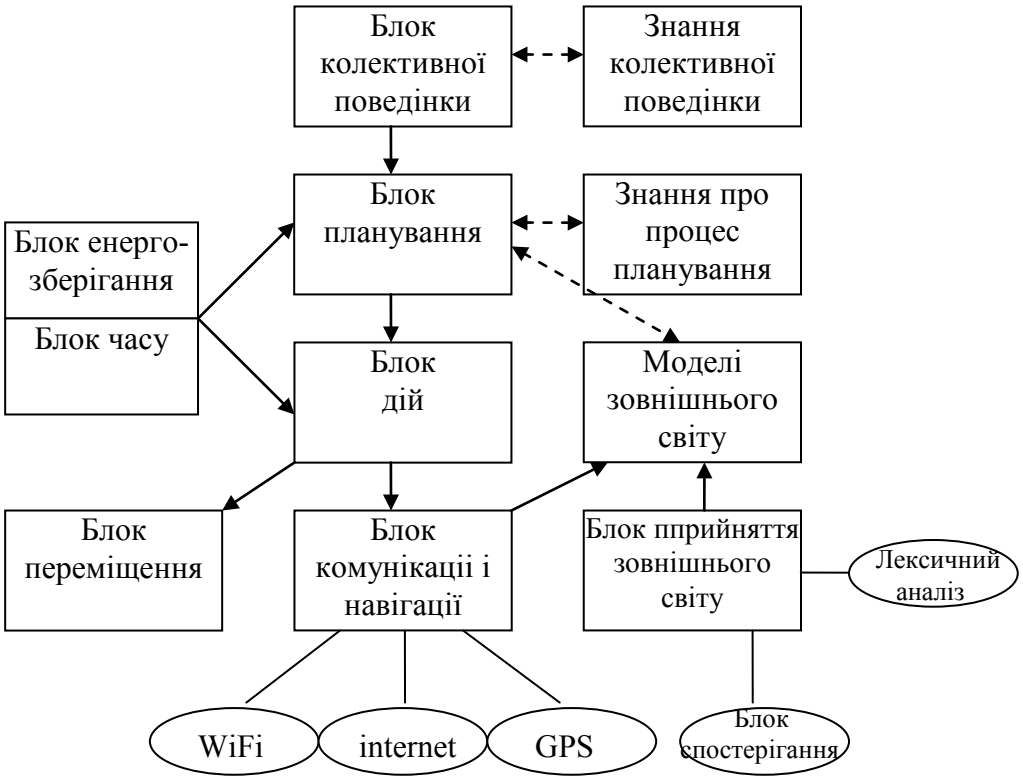

Рис. 3. Архітектура інтелектуального агента

Компонента, відповідальна за кооперацію агентів, бере участь в конструюванні планів сумісної поведінки агентів для досягнення деякої загальної мети або виконання своїх зобов'язань перед іншими агентами, а також виконання угод. Цій компоненті доступні знання всіх трьох рівнів абстракції.

База знань агента має трирівневу структуру і побудована за принципом дошки оголошень. Рівні бази знань фактично відповідають рівням абстракції знань відповідно до структури компонент. Модель світу агента містить переконання агента відповідно до рівня, орієнтованого на поведінку. Другий рівень відповідає моделі ментальних знань агента і знанням про поточний ментальний стан агента (наміри, цілі, плани). Нарешті, третій рівень містить знання і переконання агента про інших агентів, інформацію про сумісні плани, цілі і наміри, тобто те, що пов'язане з "суспільним контекстом". Усередині бази знань, як вже наголошувалося, можливий доступ з верхніх рівнів до нижніх. Наприклад, компонента, відповідальна за поведінку, не має доступу до знань про ментальну модель і до знань про кооперативну поведінку.

Загальне управління поведінкою здійснюється шляхом комунікацій між рівнями. За деякої вхідної події агент намагається розпізнати ситуацію в зовнішньому світі, і управління поступово зрушується знизу догори доти, доки не досягне рівня, здатного опанувати ситуацію, що виникла.

Очевидно, існує три варіанти реакції агента на зовнішні події:

– реакція з використанням тільки поведінкового рівня, коли цей рівень знаходить фрагмент поведінки, адекватний ситуації, без явного залучення локального планування;

– реакція з використанням локального планування, коли завдання переміщається з нижнього рівня на рівень локального планування, де і конструюється план;

– реакція з використанням рівня кооперативного планування, коли пошук плану з рівня локального планування переміщується далі на рівень планування кооперативної поведінки.

Звичайно, в цій архітектурі існують і складніші варіанти побудови плану, коли, наприклад, протокол взаємодії між рівнем локального планування і планування кооперативної поведінки передбачає складну схему обміну інформацією, наприклад, для побудови оцінок можливості розв'язання деяких задач багатоагентної системи за заданий час, через присутність в схемі системи

ситуаційного управління. Пояснимо її функціонування і особливості завдань, що вирішуються окремими підсистемами.

Блок планування

На рис. 4 зображено архітектуру блока планування інтелектуального агента. Блоки системи, названі “Аналізатор”, “Класифікатор”, “Корелятор” і “Екстраполятор”, в сукупності виконують функції блоків планування і моделювання.

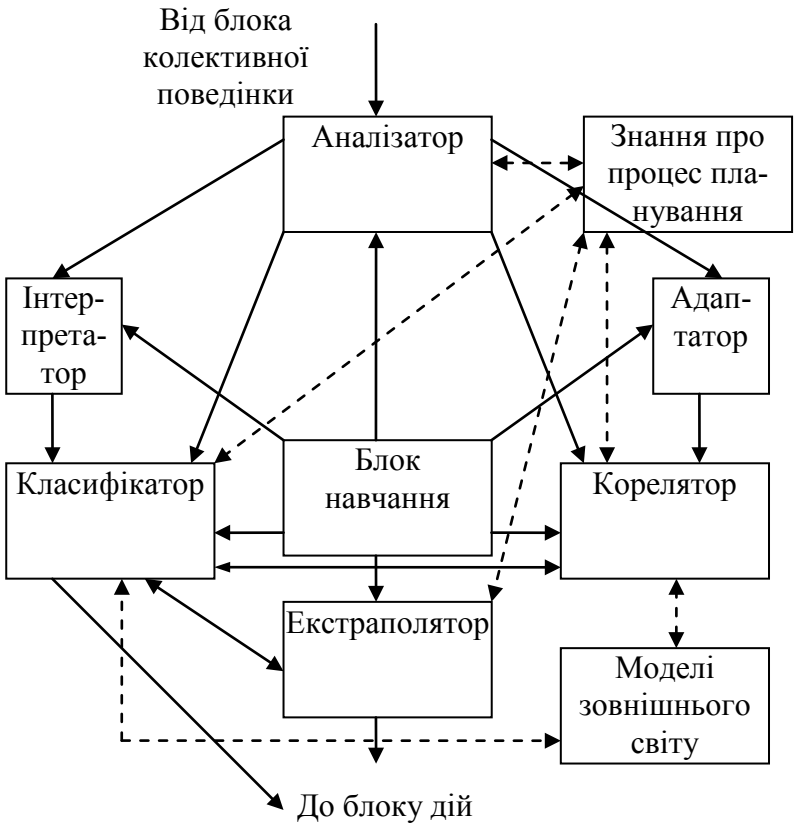

Рис. 4. Блок планування інтелектуального агента.

“Класифікатор” зберігає в своїй пам’яті відомості про ситуації, що вже відбулися. Крім того, “Класифікатор” містить в собі процедури узагальнення і конкретизації описів, а також процедури, пов’язані з функціями інформаційного банку. Під час вступу на вхід “Класифікатора” конфліктної ситуації відбувається перевірка приналежності її до узагальнених класів найвищого рівня. Якщо такий клас знайдено і цьому класу відповідає єдине рішення щодо управління, то це рішення видається безпосередньо на блок дій. Якщо ж для тієї конфліктної ситуації, що надійшла в “Класифікатор”, знаходиться такий узагальнений опис, для якого є альтернативою вибір рішення з управління, або ситуація, що надійшла, взагалі не входить ні до якого з наявних в “Класифікаторі” узагальнених описів, то, перш ніж видати рішення, “Класифікатор” звертається до інших блоків. За наявності декількох альтернативних рішень остаточне рішення про вибір керівної дії переходить до “Екстраполятора”.

“Екстраполятор” містить набір процедур, серед яких є процедури дедуктивного і індуктивного висновків. Крім того, в “Екстраполяторі” можуть міститися і звичайні імітаційні процедури, за допомогою яких відбувається “програвання” наслідків ухвалення тих або інших рішень на декілька кроків вперед. Процедури планування дають змогу розв’язувати дві задачі: будувати багатокрокове управління й оцінювати ефективність тих або інших альтернативних рішень. Після вибору керівної, або плану багатокрокової дії “Екстраполятор” видає її на блок дій і одночасно

повідомляє про свій вибір "Класифікатору", який використовує цю інформацію для поліпшення даних про пройдені ситуації.

Найважчим для управління є той випадок, коли ситуація, що надійшла, ніяк не класифікується в "Класифікаторі". У цьому випадку інформація про неї надходить у "Корелятор", в якому зберігаються знання про закони функціонування об'єкта, обмеження на керівні дії, що подаються на об'єкт, і цільові структури. На підставі інформації, що надійшла, "Корелятор" вибирає допустиму дію на об'єкт управління і видає інформацію про цей вибір в "Класифікатор". "Класифікатор" за цим рішенням відносить спостережувану ситуацію, що надійшла, до тих узагальнених описів, яким відповідає рішення, сформоване в "Кореляторі". Якщо це узагальнення не породжує інших альтернативних рішень, то "Класифікатор" видає інформацію про ухвалене рішення в блок дій. Якщо ж в результаті узагальнення виявиться, що в спостережуваній ситуації можна ухвалити й інші рішення, відмінні від рекомендованого "Корелятором", то ці рішення повідомляють останньому. "Корелятор" перевіряє допустимість нових рішень. Всі допустимі рішення потім передаються в "Екстраполятор", який і вибирає серед них остаточне рішення щодо управління.

Було описано процес функціонування системи ситуаційного управління на етапі її роботи. Проте, як і для всіх відкритих систем управління, цьому етапу передує етап початкового навчання системи. Основними процедурами цього етапу є процедура побудови ситуації, що вже відбулася в "Класифікаторі", і процедура побудови системи правил висновку в "Кореляторі". На цьому етапі система ситуаційного управління працює в режимі діалогу з проектувальниками системи і фахівцями-технологами. Цей діалог здійснюється через спеціальний блок навчання. Через цей самий блок відбувається первинне заповнення всіх основних блоків апріорною інформацією, яку можна вкласти в систему управління і без навчання. Процес навчання продовжується весь час (частково за рахунок тих процедур, які пов'язані з деформацією ситуації, що вже відбулися внаслідок діяльності блоків системи в процесі роботи).

Як і в будь-якій системі управління, побудованій за типом систем ситуаційного управління, обов'язково повинен бути присутнім блок, в якому зберігається модель знань про зовнішній світ. У цій моделі знань повинні зберігатися описи ситуацій, що складаються на об'єкті управління, процедури узагальнення цих описів, схеми ухвалення рішень з управління і процедури їх адаптації в реальних умовах, дані про обмеження, властиві об'єкту з погляду управління ним тощо. Іншими словами, ця модель знань повинна реалізувати функції як «Корелятора», так і «Класифікатора». Такою ж важливою є наявність процедур за визначенням конфліктності ситуації, незалежно від того, де і в якому з блоків системи ці процедури будуть реалізовані. Саме сукупність вказаних процедур і обумовлює специфіку систем ситуаційного управління.

Висновки

Запропоновано вирішення проблеми обміну знаннями між окремими модулями архітектури інтелектуального агента з врахуванням колективної поведінки агента. Розроблено гібридну архітектуру інтелектуального агента для автономних мобільних роботів з алгоритмами узгодження його рівнів, а також запропоновано механізм взаємодії блоку планування інтелектуального агента, що може дати змогу агенту виконувати невідомі завдання ускладненого типу. Порівняно існуючі архітектури агента, проаналізовано і класифіковано існуючі архітектури інтелектуального агента.

1. Поспелов Д.А. Логико-лингвистические модели в системах управления. – М.: Энергоиздат, 1981. – 232 с. 2. Walker A. and Woodridge M. Understanding the emergence of Conventions in Multi-Agent Systems. In Proceedings. – 1995. 3. Muller J.P., Pishel M., and Thiel M. Modelling Reactive Behaviour in Vertically Layered Agent Architectures. In: Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. Amsterdam, The Netherlands, August 8–9, 1994; Eds. M.J. Wooldridge and N.R. Jennings. Proceedings. Springer Verlag. – 1994. – P. 261–276. 4. Moffat D. and Frijda N.H. Where there's a Will there's an Agent. n: Intelligent Agents. ECAI-94 Workshop on Agent

Theories, Architecture and Languages. Amsterdam, The Netherlands, August 8–9, 1994; Eds. M.J. Wooldridge and N.R. Jennings). Proceedings. Springer Verlag. – 1994.– P. 245–259. 5. Ferguson I.A. Integrated Control and Coordinated Behaviour: A case for Agent Models. In: Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. 6. R.E.Fikes and N.Nilsson. STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence, 5(2). – 1971. – P. 189–208. 7. Wooldridge M. and Jennings N.R. Agent Theories, Architectures, and Languages: A Survey. In: Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. Amsterdam, The Netherlands, August 8–9, 1994; Eds. M.J. Wooldridge and N.R. Jennings. Proceedings. Springer Verlag. – 1994. – P. 3–39. 8. Dunin-Keplicz B. and Treuer J. Compositional Formal Specification of Multi-Agent System In:Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. Amsterdam, The Netherlands, August 8–9, 1994; Eds. M.J. Wooldridge and N.R. Jennings. Proceedings. Springer Verlag. – 1994.– P. 102–117. 9. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы // Новости искусственного интеллекта. – 1997 – № 1. 10. Тарасов В.Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // Новости искусственного интеллекта. – 1998. – № 2. – С. 5–63. 11. Городецкий В. И. Многоагентные системы: современное состояние исследований и перспективы применения // Труды конференции по ИИ. – 1996. – С. 36–45.

УДК 004.31, 004.056.55, 003.26

Л.М. Коркішко
Тернопільський державний економічний університет,
кафедра безпеки інформаційних технологій

БАЗОВІ ЛОГІЧНІ ЕЛЕМЕНТИ ДЛЯ КОМП'ЮТЕРНИХ ПРИСТРОЇВ ЗАХИСТУ ІНФОРМАЦІЇ

Запропоновано узагальнений параметризований метод для побудови базових логічних елементів (логічного множення та додавання), призначених для використання у комп'ютерних пристроях захисту інформації.

It is proposed a generalized parameterized method for creation of basic logical elements (logical AND and logical OR) intended for usage in computer devices for information protection.

Вступ

Із розширенням галузей застосування криптографічних перетворень у сучасному житті (банківські транзакції, смарт-карти, персональні комунікаційні пристрої тощо) значно зростає роль конфіденційності даних. Компрометування цих даних (наприклад, отримання зловмисником відомостей про ці дані) створює можливість реалізації загроз безпеки для їх власника, наприклад, його фінансових втрат. Для отримання відомостей про конфіденційні дані, які використовуються у криптографічних перетвореннях, зловмисник може використати інженерно-криптографічні атаки за побічними каналами витоку інформації [1–8]. Комп'ютерні пристрої захисту інформації уможливлюють витік інформації через сигнал про споживану потужність під час криптографічних перетворень з використанням конфіденційних даних (ключів шифрування чи цифрового підпису).

Для отримання відомостей про використовувані конфіденційні дані з сигналу про споживану потужність комп'ютерного пристрою використовують спеціальні методи аналізу, так званий "диференційний аналіз споживаної потужності" (ДАСП) [8]. Ці методи аналізу характеризуються