

В.М. Єфремов, аспір.

Інститут кібернетики НАН України

ПОБУДОВА СИСТЕМ УПРАВЛІННЯ АВТОГОСПОДАРСТВОМ ІЗ ЗАСТОСУВАННЯМ ПІДХОДУ БАГАТОАГЕНТНОГО ПРОГРАМУВАННЯ

Розглядаються основні принципи, які входять в основу розробки і програмної реалізації комплексного інструментарію для побудови спеціалізованих експертних систем автогосподарства з врахуванням системно-організаційного підходу багатоагентного програмування. Пропонуються інструментальні засоби для побудови таких систем.

Класична парадигма багатоагентного програмування (БП), яка базується на індивідуальному підході до характеру інтелекту програмної системи, пропонує формування чіткого моделювання специфічних аспектів інтелекту (наприклад, роботи з базами знань). Отже, це накладає деякі принципові обмеження при побудові інтелектуальних задач.

Виходячи з парадигм БП, системний підхід протиставлений логічному підходу. Серед основних блоків у сучасному БП концепція агента повинна мати узгоджений інтерес для обміну інформацією з іншими агентами. Різні формування чіткості агентів строго обумовлені та класифіковані, загальна модель дії запропонована як ядро для вибору архітектури агента.

Теорія штучного інтелекту, яка активно розробляється в останні десятиліття багатьма дослідниками, почала знаходити все ширше практичне застосування при побудові програмних систем, які працюють не тільки з традиційними і первинними даними, але і з базами знань, які відносяться до конкретної проблемної області задачі. Використання представлених у зручній для користувачької та машинної обробки елементів знань дозволило розширити коло задач, які вирішуються із застосуванням ЕОМ в автоматизованому і автоматичному режимах. При цьому, включивши до їх складу задачі з неявно вираженим завданням, оперують як кількісними, так і якісними характеристиками, які потребують моделювання людських роздумів тощо.

Ідея інтелектуального посередника «агента» виникла у зв'язку з бажанням спростити стиль спілкування кінцевого користувача з програмами, оскільки домінуючий на сьогодні стиль взаємодії користувача з комп'ютером припускає, що користувач запускає задачу і керує її рішенням. Але це зовсім не підходить для недосвідченого користувача у випадку необхідності внесення змін у програму. Інакше кажучи, спочатку ідея інтелектуального посередника в системі «людина – ЕОМ – людина» виникла як спроба інтелектуалізації користувачького інтерфейсу.

Застосування методів штучного інтелекту дозволило зробити перший крок до зміни стилю взаємодії користувача з комп'ютером. Виникла ідея створення так званих «автономних агентів», які визначили уже новий стиль взаємодії користувача з програмою. Замість традиційної взаємодії, яка ініціюється користувачем шляхом команд і прямих маніпуляцій, користувач втягується у сумісний процес рішення. При цьому як користувач, так і комп'ютерний посередник – «інтелектуальний агент» (ІА), обидва беруть участь у запуску задачі, управлінні подіями і вирішенні задачі.

Для такого стилю використовується метафора «персональний асистент» (ПА), який співробітничує з користувачем в тому ж робочому середовищі.

Необхідність розгляду основних задач автоматизації автогосподарства потребує комплексного підходу до вирішення управлінських, фінансових, аналітичних, інформаційних задач з врахуванням суттєвого впливу на його структуру і методи управління різного роду ринкових факторів, наприклад, таких як зміни у законодавчих актах, зміни ринку споживачів і постачальників тощо. Це потребує специфічного підходу до побудови систем автоматизації автогосподарством. Такі системи на сьогоднішній день повинні мати наступні специфічні особливості в порівнянні із створеними раніше:

- універсальність програмного продукту, який дозволяє автоматизувати управління автогосподарств з різною структурою;
- масштабування з точки зору нарощування кількості та збіжності програм;
- можливість настройки системи на різні види розрахунків, форм введення даних, вихідних звітів тощо;
- можливість взаємодії з іншими видами систем, наприклад, «К/З», «Бухгалтерія 1С» та іншими;
- використання сучасних СУБД, інтерфейсних засобів людино-машинної взаємодії (в тому числі графічних) та операційних систем, які забезпечують як індивідуальну роботу системи, так і роботу у мережному варіанті.

З врахуванням перерахованих вище умов була розроблена автоматизована система управління автогосподарством «Агенти автопідприємства».

До складу запропонованої системи входять агенти для наступних задач:

- розрахунок заробітної платні;
- облік надходжень палива;
- облік витрат палива;
- виписка, облік і реєстрація шляхових листів;
- розрахунок маршрутів згідно заявок;
- облік основних фондів;
- облік матеріальних цінностей;
- виписка платіжних доручень;
- виписка податкових накладних;
- спеціалізована програма по створенню та реєстрації документів, листів, актів, вимог тощо.

Система може працювати як під керуванням операційної системи DOS, так і у Windows.

Робота системи. Більшість програм, розроблених раніше, працює за принципом вибору із списку запропонованих послуг, які надаються програмою, а потім – вибору необхідних дій даної програми тощо.

Робота системи «Агенти автопідприємства» побудована на основі роботи з агентами. Кожен користувач, який працює з даною системою має свій пароль доступу і одночасно отримує програмного агента, або як його називають персонального асистента (ПА), який запам'ятовує послідовність роботи користувача, і запам'ятовує їх у своїх властивостях.

Агент системи «Агенти автопідприємства» формує свою компетенцію з чотирьох різних дже-

рел.

Перше джерело – агент навчається шляхом постійного «підглядання із-за плеча користувача», коли користувач виконує деякі дії. Агент може відслідковувати дії користувача, пам'ятати траєкторії всіх його дій за тривалий період (при необхідності за тижні і навіть місяці), відслідковувати закономірності та фрагменти, які повторюються, і пропонувати способи їх автоматизації. Так, наприклад, якщо агент помітив, що користувач протягом деякого часу, працюючи з однією і тією ж програмою, постійно використовує дію (запис поточних даних), то він запам'ятовує це, а потім може запропонувати користувачу автоматизувати цю дію.

Друге джерело для навчання – це прямий і непрямий зворотний зв'язок від користувача. Непрямий зворотний зв'язок має місце тоді, коли користувач нехтує пропозиціями агента і замість цього вибирає альтернативну дію. Це може бути слушним, коли користувач змінює порядок, в якому він планує працювати, ігноруючи пропозицію, подану агентом, або корегує порядок настрійок, які запропонував агент. Користувач може також відмінити дії, автоматизовані агентом, наприклад "Не виконувати запуск даної програми", "Не складати звіт" та інші.

Третє джерело – це інформація у вигляді прикладів, які задаються користувачем у явному вигляді. Користувач може «тренувати» агента, видаючи гіпотетичні приклади подій та ситуацій і повідомляючи агента, що робити в кожному конкретному випадку. Інтерфейсний агент запам'ятовує цю дію, формує відношення між об'єктами і змінює свою базу прикладів так, щоб узгодити її із заданим прикладом. Наприклад, користувач може навчати агента розподілу проведення звітів, вказуючи строго визначену послідовність, шляхом створення гіпотетичного прикладу – плану (який не описаний у всіх аспектах, крім списку робочих програм і зв'язків між ними) і перетягувати це повідомлення у множину прикладів.

Нарешті, четверте джерело, яке використовується агентом для формування компетентності, – це запит «поради» у іншого агента, який асистує іншому користувачу з аналогічною задачею (і який може бути побудований на основі «більшого» досвіду). Якщо агент не представляє сам, яка дія є придатною в конкретній ситуації, він може представити ситуацію іншому агенту і «запитати», «яку дію він рекомендує для даної ситуації». Наприклад, якщо прибула заявка на доставку вантажу в пункт Х, який раніше не зустрічався у списку прикладів агента, то агент може запросити в інших агентів, що робити з цією заявкою. В цьому випадку, якщо буде виявлений агент, який оброблявся раніше подібні дії, він зможе надати список всіх необхідних дій для обробки даної ситуації. (Тут не розглядається ситуація з перевіркою прав на проведення даних дій). Однак в більш розгорнутому вигляді це відбувається наступним чином. Якщо більшість інших агентів рекомендують вважати його повідомленням найвищого пріоритету, яке повинно бути заплановане в першу чергу, то агент може запропонувати цю рекомендацію своєму користувачу, навіть якщо раніше він ніколи не спостерігав, щоб користувач мав справу з пунктом Х. Скоріш за все замість рекомендацій всіх агентів користувач може інформувати свого агента сприймати рекомендації з боку одного або декількох специфічних агентів, які асистують конкретним користувачам. Наприклад, якщо один із користувачів системи є експертом з деякого виду робіт (наприклад, формує маршрути перевезень), то інші користувачі можуть інструктувати своїх агентів, приймати поради про перевезення від агента користувача – експерта. Для реалізації даної системи була використана багаторівнева архітектура для розподілених прикладних про-

грам. Вона включає багаторівневу структуру знань, робочу пам'ять, менеджера комунікацій та людино-машиний інтерфейс (рис. 1).

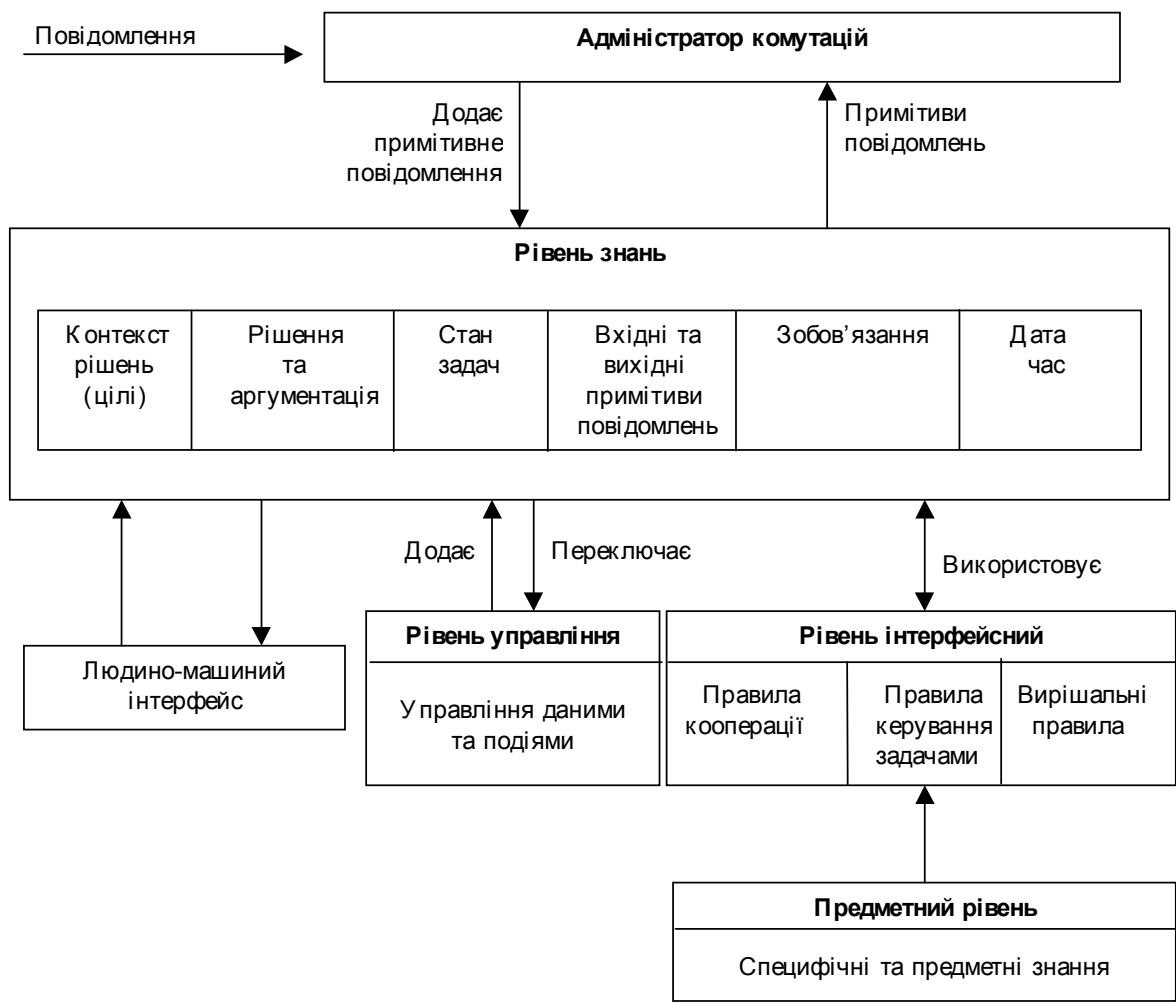


Рис. 1

Оскільки дана архітектура повинна бути релевантною прикладним програмам роботи з автопідприємствами, агент повинен володіти обома типами поведінки: поведінкою на основі знань, наприклад, для вибору планів, декомпозиції задач, розміщення задач та поведінкою на основі швидкої реакції на події (наприклад, для формування відповідей в реальному часі на нові дані, які поступають, модифікація даних, які є, зміни поточних домовленостей з іншими агентами). Таким чином, ця архітектура є гібридною.

В даній системі інтелектуальна поведінка агентів підтримується спільною роботою таких компонент, як компонент вирішальних правил для обчислення плану, компонент правил для керування задачами, їх декомпозицією та розміщенням, а також компонент правил для підтримки домовленостей з іншими агентами при кооперативному рішенні задач.

Реактивна поведінка реалізується за допомогою керуючого рівня, який реагує на зміни стану робочої пам'яті (наприклад, при надходженні нових результатів рішення задачі, цілей або повідомлень, а також при модифікації баз даних, цілей, міжагентських домовленостей або станів за-

дач). Ключовим моментом даної архітектури є трирівнева організація знань, при цьому виділяються наступні рівні:

1. Рівень специфічних предметних знань, в якому містяться маршрути руху, поточні плани ремонту автотранспорту, база даних про персонал автопідприємства і база даних про доступні ресурси автопідприємства. Однак предметні знання не містять якої-небудь інформації про те, як їх необхідно використовувати, тут представлені тільки властивості предметної області.

2. Рівень знань про процедури виведення, який містить декларативні правила виведення, що повинні застосовуватися до предметних знань в кожному конкретному випадку (рентабельність заявки, вибір маршруту, суміщення маршрутів, поточний ремонт), щоб вивести нові дані. Цей рівень – основний в архітектурі. В свою чергу, він підрозділяється на компоненти прийняття рішень в умовах невизначеності, управління задачами та управління кооперацією агентів. Наприклад, модуль управління задачами містить декларативну схему виведення для управління переходами станів задач. Особливості системи виведення рішень полягають в тому, що вона не використовує поняття ментального стану агента (переконання, бажання, наміри) і не використовує яку-небудь логічну мову для виведення, для цього вона використовує стратегії аргументації в умовах невизначеності. Це означає, що дана архітектура не є BDI-архітектурою. Менеджер задач відповідальний за декомпозицію задач на підзадачі та їх розподіл за відповідними агентами, а також за управління переходами станів задач. Управління кооперацією агентів використовує механізм, який базується на взаємних зобов'язаннях агентів ("будь-який агент згоден розпочинати схему дії, яка має за мету виконати задачу за придатний час") і домовленостях про те, при яких умовах агент має право відмовитися від своїх зобов'язань і як він повинен себе вести по відношенню до інших агентів, коли такі обставини виникнуть.

3. Рівень керуючих знань, який використовує знання про процес виведення до предметних знань, щоб генерувати схему виведення, якщо в робочу пам'ять додаються нові знання.

Такий функціональний розподіл знань на предметні знання, знання про процедури виведення та керуючі знання суттєво спрощує їх подання всередині компонент, а також їх повторне використання та експлуатацію, оскільки такі компоненти можуть створюватися і підтримуватися незалежно. Крім того, ця архітектура дозволяє просто вмонтовувати програми витягу знань, кожна з компонент яких може отримуватися та модифікуватися незалежно одна від одної.

Інші три компоненти архітектури, що розглядається, – робоча пам'ять, менеджер комунікацій та людино-машинний інтерфейс.

Робоча пам'ять служить для запам'ятовування поточних даних, які генеруються рівнем управління, користувача і менеджера комунікацій. Типи інформації, яка зберігається в робочій пам'яті, такі: цілі, які повинні бути досягнуті; стани задач, які знаходяться в поточному стані процесу виконання домовленостей з іншими агентами. Фактично, у звичній нам термінології, робоча пам'ять є ніщо інше, як дошка об'яв.

Менеджер комунікацій містить повідомлення, які повинні бути послані іншим агентам, представлені на мові комунікацій з примітивами типу примітивів мови KQML: звернутися з проханням, прийняти, заперечити, змінити, запропонувати, проінформувати, запитати дані, відмовитися та підтвердити.

Людино-машинний інтерфейс визначає схему взаємодії між системою та користувачем, оскільки дана багатоагентна система не є автономною.

Ця архітектура базується на знаннях, має горизонтальну схему взаємодії рівнів. Головна її особливість в тому, що вона достатньо сильно орієнтована на прикладні програми.

ЛИТЕРАТУРА:

1. *Городецкий В.И., Грушинский М.С., Хабалов А.В.* Многоагентные системы // Санкт-Петербургский Институт Информатики и автоматизации РАН
2. *Тарасов В.Б.* О системно-организационном подходе в искусственном интеллекте // Санкт-Петербургский Институт Информатики и автоматизации РАН
3. *Городецкий В.И.* Многоагентные системы: современное состояние исследований и перспективы применения // КИА 96. Том 1
4. *Бондарев П.А., Силкин В.В.* Программная реализация инструментальных средств для построения экспертных систем // Информационные технологии. – 1998. – № 9.

ЄФРЕМОВ Віталій Миколайович – аспірант Інституту кібернетики НАН України.

Наукові інтереси:

– дослідження та розробка системи засобів складального програмування.

Подано 21.12.1999.