

Програмування інтелектуальних інформаційних систем

3 курс, осінь 2021

- Доц. Баклан І.В.
- Email: iaa@ukr.net
- Web: baklaniv.at.ua

Лекції 1-2.
ІНТЕЛЕКТУАЛЬНІ
СИСТЕМИ

Вступ до дисципліни

У XXI ст. безліч професій пов'язано з інтенсивною багатофункціональною переробкою різних видів інформації, поступово більшість людей відчуває стрес - людство захлинається в інформаційному морі. Комп'ютер і особисті мобільні гаджети могли б стати помічником, але часто стають додатковим джерелом зайвої і непотрібної інформації. Фахівці потребують інтелектуального помічника, який би зміг відфільтрувати, узагальнити і стиснути інформаційний потік. Тому “Програмування інтелектуальних інформаційних систем” на сьогодні набуває актуальності в межах підготовки студентів за спеціальністю “Інженерія програмних систем”.

Інженерія знань - перша наука про способи компресії інформації. На її основі можливо тиражування знань через інтелектуальні системи, які покликані допомогти фахівцям в обробці складної інформації, прийнятті рішень, аналізі та інтерпретації даних. Для створення будь-якої комп'ютерної системи необхідний АНАЛІТИК - людина, яка попередньо проаналізує проблему і поставить задачу машині. Але і крім інтелектуальної системи інженерія знань допомагає зробити знання видимими, це потрібно для кращого розуміння проблем і широко використовується в провідних компаніях для управління корпоративними знаннями.

Під аналітиками будемо розуміти як системних аналітиків і архітекторів, які проектують програмні системи, так і бізнес-аналітиків, що ставлять для них завдання. Так склалося, що першими про проблемі структурування знань задумалися розробники інтелектуальних систем. Але сьогодні інженерія знань переступила рамки розробки інтелектуальних систем і перетворилася в міждисциплінарну область, що досліджує проблеми вилучення, структурування, подання та обробки знань. Вирішення цих задач тісно переплітається з різними науковими напрямками менеджменту і маркетингу, психології та філософії, лінгвістики і математики, біології та кібернетики.

Дисципліна “Програмування інтелектуальних інформаційних систем” розрахована на майбутніх розробників систем штучного інтелекту, інтелектуальних систем, баз знань, систем управління базами знань, експертних систем. Акцент ставиться на нових візуальних моделях структурування знань (інтелект-карти, концептуальні графи і ін.), які служать потужним інструментом аналізу складних предметних областей. Зараз вже нікому не потрібно доводити актуальність систем, заснованих на знаннях, більш того, знання стали модним прапором, під яким стрімко набирає оберти досить молода наука - управління знаннями. Проблематика опрацювання знань стає центральною в сучасному менеджменті. Фахівці йдуть, несучи з собою свій інтелектуальний капітал. Управління знаннями є сукупністю процесів, які керують створенням, розповсюдженням, обробкою і використанням знань всередині підприємства, організації.

Особливо глибоко і детально розглядається повний цикл розробки інтелектуальною системою тиражування знань - від вилучення і формалізації знань до реалізації компонентів прикладної інтелектуальної системи.

При цьому істотний акцент робиться саме на інженерії знань і онтологічному інжинірингу, новому розділу інженерії знань, орієнтованому на розробку концептуальних моделей будь-яких складних систем від держави до людського організму. Ці моделі створюються саме для того, щоб моделювати і тиражувати знання, якими володіють люди-експерти.

При цьому питання подання знань в пам'яті людини і процес міркувань і на сьогодні залишаються мало вивченими. Уявний на перший погляд всесильний апарат логічних міркувань не може дати прийняттого рішення для цілого класу проблем. Багато в чому це пояснюється тим, що логіка традиційно займалася так званими строгими міркуваннями, залишаючи за бортом інші багаточисельні форми людських міркувань. Уявлення про модель пам'яті людини в психології дуже громіздкі і мало що дають для побудови схеми міркувань людини в конкретних ситуаціях.

В лекціях зроблена певна спроба максимально широко описати різні аспекти проектування та програмування баз знань і систем управління знаннями — інтелектуальних інформаційних систем. Послідовно і детально розглядаються питання, які необхідно вирішити розробнику в процесі створення нової системи. Наводиться опис можливих альтернативних рішень, із зазначенням області застосування тих чи інших підходів, дається історична довідка.

1.1. Стислий історичний екскурс

Загадки феномена мислення і життя хвилювали людство з давніх часів. Так, ідея створення штучного подібності людини з елементами «розумності» сходить до єгиптянам, які в III столітті до н. е. створили «оживаючу» механічну статую бога Амона, яка видає за запитом «передбачення» майбутнього.

У давньогрецькій міфології скульптор Пігмаліон, цар Кіпру, зробив із слонової кістки статую прекрасної жінки й закохався в неї. На його благання Афродіта оживила статую, яка дістала ім'я Галатея і стала дружиною митця.

У Гомера в «Іліаді» бог Гефест кував людиноподібні істоти-автомати.

Сократ мріяв про механічного «безпристрасного» суддю.

Ґо́лем — персонаж єврейської міфології, легендарна людиноподібна істота, штучно створена рабинами-кабалістами у Празі. Глиняний велетень буцімто міг виконувати певні дії після «програмування» його кабалістичними заклинаннями. При «збої» такої «програми» Ґо́лем трошив все навколо.

Мері Шеллі у своєму романі «Франкенштайн, або Сучасний Прометей» розповідає про молодого студента-науковця, який внаслідок загадкового наукового експерименту створює гротескну, але чутливу істоту.

Передісторія

Родоначальником штучного інтелекту вважається середньовічний іспанський філософ, математик і поет *Раймунд Луллій*, який ще в XIII в. понамагався створити механічну машину для вирішення різних завдань на основі розробленої ним загальної класифікації понять. Луллій описував свої спроби механічно моделювати логічні операції у винайденій ним із цією метою «логічній машині», аналізував логічний характер питальних речень, вивчав взаємини логічних констант «і» та «або» (vel). Формальну логіку Луллій визначав як «мистецтво і науку, за допомогою яких істина і неправда розпізнаються розумом і відокремлюються одна від одної, — наука прийняття істини і відбраковування неправди». У наведеному визначенні міститься явна вказівка на двозначний характер логічних формалізмів. У «Великому мистецтві» Луллій описав також і техніку моделювання логічних операцій — систему концентричних кіл, з яких кожне слугує для позначення групи понять.

У XVIII ст. великі математики *Лейбніц* і *Декарт* незалежно один від одного продовжили цю ідею, запропонувавши універсальні мови класифікації всіх наук. Ці роботи можна вважати першими теоретичними роботами в області штучного інтелекту. Однак тільки винайдений Чарльзом Беббідж (Charles Babbage) в 1833 р механічний калькулятор може вважатися прямим предком інтелектуального комп'ютера. Його помічник, леді Ада Лавлейс (Lady Lovelace), племінниця Байрона, стала першим користувачем і програмістом.

Стисла історія

Народження штучного інтелекту як наукового напрямку відбулося тільки після створення ЕОМ в 40-х рр. ХХ ст. В цей же час Норберт Вінер створив свої основні роботи по новій науці - кібернетиці.

Термін «штучний інтелект» - П (artificial intelligence - AI) - був запропонований в 1956 р в рамках річного проекту з аналогічною назвою в Дартмутском коледжі (США). Проект був розрахований на 2 місяці, в ньому приймали участь 10 осіб. Тема семінару - розробка логічних, а не обчислювальних завдань. В англійській мові дане словосполучення не має тієї злегка фантастичного антропоморфної забарвлення, якого воно набуло в досить невдалому перекладі. Слово *intelligence* означає «уміння міркувати розумно», а зовсім не «інтелект», для якого є термін *intellect*.

Варто докладніше розповісти про Дартсмутском семінарі, один список учасників якого вже увійшов в історію ШІ, як легендарний: Джон Маккарті (John McCarthy) - батько мови ЛІСП (LISP) і глава лабораторії ШІ в Стенфордському університеті, Марвін Мінський (Marvin Minsky) - піонер ШІ і автор ідеї фрейму, Клод Шеннон (Claude Shannon) - лауреат Нобелівської премії, засновник теорії інформації. Семінар розпочався з обговорення програми «Логік-теоретик» для автоматичного доведення теорем, причому одна з теорем несподівано була доведена новим покращеним способом. До кінця другого місяця ідея ШІ була сформульована, амбіції і ентузіазм учасників були цілеспрямовані навчити машину здійснювати деякі «розумні», осмислені, а не просто обчислювальні процедури, наприклад, грати в шахи, переводити на іноземні мови або складати музику. Дещо з цього збулося.

Незабаром після визнання штучного інтелекту окремою галуззю науки відбувся поділ його на два напрямки: *нейрокібернетика* і *кібернетика* «Чорного ящика». Ці напрямки розвиваються практично незалежно, суттєво розрізняючи як в методології, так і в технології. І тільки в 90-і рр. Стали помітні тенденції до об'єднання цих частин знову в єдине ціле.

Основну ідею нейрокібернетики можна сформулювати наступним чином: єдиний об'єкт, здатний мислити, - це людський мозок, тому будь-який «Мислячий» пристрій повинен якимось чином відтворювати його структуру.

Таким чином, нейрокібернетика орієнтована на програмно-апаратний моделювання структур, подібних структурі мозку. Фізіологами давно встановлено, що основою людського мозку є велика кількість (до 10^{21}) пов'язаних між собою і взаємодіючих нервових клітин - нейронів. Тому зусилля нейрокібернетики були зосереджені на створенні елементів, аналогічних нейронам, і їх об'єднання в функціонуючі системи. Ці системи прийнято називати *нейронними мережами*, або *нейромережами*.

Перші нейромережі були створені *Розенблатта* і *Мак-Каллоком* в 1956-1965 рр. Це були спроби створити системи, що моделюють людське око і його взаємодію з мозком. Пристрій, створений ними тоді, отримало назву *перцептрона (perceptron)* . Воно вміло розрізняти букви алфавіту, але було чутливо до їх написання. Наприклад, букви **A** , **A** і *A* для цього пристрою були трьома різними знаками. Поступово в 70-80-х рр. кількість робіт в цьому напрямку штучного інтелекту стало знижуватися. Занадто невтішні були перші результати. Автори пояснювали невдачі малої пам'яттю і низьким швидкодією існуючих в той час комп'ютерів.

Однак в 1980-х рр. в Японії в межах проекту «ЕОМ V покоління» був створений перший *нейрокомп'ютер* , або комп'ютер VI покоління. До цього часу обмеження по пам'яті і швидкодії були практично зняті. З'явилися *трансп'ютери* - паралельні комп'ютери з великою кількістю процесорів.

Трансп'ютерна технологія - це тільки один з десятка нових підходів до апаратної реалізації нейромереж, які моделюють ієрархічну структуру мозку людини. Основна область застосування нейрокомп'ютерів - розпізнавання образів.

В цілому, можна виділити 3 підходи до створення нейромереж:

1. *Апаратний* - створення спеціальних комп'ютерів, нейрочипів, плат розширення, наборів мікросхем, що реалізують всі необхідні алгоритми.
2. *Програмний* - створення програм і інструментаріїв, розрахованих на високопродуктивні комп'ютери. Мережі створюються в пам'яті комп'ютера, всю роботу виконують його власні процесори.
3. *Гібридний* - комбінація перших двох. Частина обчислень виконують спеціальні плати розширення (співпроцесори), частина - програмні засоби.

В основу підходу, названого кібернетикою «чорного ящика», було покладено принцип, протилежний нейрокібернетиці: «Не має значення, як устроєний "мислячий" пристрій. Головне, щоб на задані вхідні дані воно реагувало так само, як людський мозок».

Прихильники цього напрямку мотивували свій підхід тим, що людина не повинна сліпо слідувати природі в своїх наукових і технологічних пошуках. Так, наприклад, очевидний успіх колеса, якого не існує в природі, або літаків, що не махають крилами, наслідуючи птиці. До того ж прикордонні науки про людину не змогли внести істотного теоретичного вкладу, об'ясняючого хоча б приблизно, як протікають інтелектуальні процеси у людини, як влаштована пам'ять і як людина пізнає навколишній світ.

Даний напрямок штучного інтелекту було орієнтоване на позови алгоритмів вирішення інтелектуальних завдань на існуючих моделях комп'ютерів. Істотний внесок у становлення нової науки внесли її «Піонери»: Маккарті (вже згаданий автор першої мови програмування для задач ШІ - ЛІСП), Мінський (автор ідеї *фрейма* і *фреймової* моделі подання знань), Ньюелл, Саймон, Шоу, Хант і інші.

У 1956-1963 рр. велися інтенсивні пошуки моделей і алгоритмів людського мислення і розробка перших програм на їх основі. представники існуючих гуманітарних наук - філософи, психологи, лінгвісти - ні тоді, ні зараз не могли запропонувати таких алгоритмів. Тоді кібернетики почали створювати власні моделі. Послідовно були створені і випробувані різноманітні підходи.

В кінці 50-х рр. народилася модель *лабіринтового пошуку*. Це підхід подання завдання як деякий простір станів у формі графа, і в цьому графі проводиться пошук оптимального шляху від вхідних даних до результуючих. Була проведена велика робота по розробці цієї моделі, але в рішенні практичних завдань ідея великого поширення не отримала. У перших підручниках з штучного інтелекту [Хант, 1986; Ендрю, 1985] описані ці програми - вони грають в гру «П'ятнашки», збирають «Ханойську вежу», грають в шашки і шахи.

Початок 60-х рр. - це епоха *евристичного програмування*. *Евристика* - правило, теоретично не обґрунтоване, яке дозволяє скоротити кількість переборів в просторі пошуку. Евристичне програмування - розробка стратегії дій на основі відомих, заздалегідь заданих евристик.

У 1963-1970-х рр. до вирішення завдань стали підключати методи *математичної логіки*. Робінсон розробив *метод резолюцій*, який дозволяє автоматично доводити теореми при наявності набору вихідних аксіом. Приблизно в цей же час видатний радянський математик Ю. С. Маслов запропонував так званий *зворотний висновок*, згодом названий його ім'ям, розв'язуючий аналогічне завдання іншим способом. На основі методу резолюцій француз Альбер Кольмерое в 1973 р створює мову логічного програмування *Пролог*. Великий резонанс мала програма «*Логік-теоретик*», створена Ньюелом, Саймоном і Шоу, яка доводила шкільні теореми.

Однак більшість реальних завдань не зводиться до набору аксіом, і людина, вирішуючи виробничі завдання, не використовує класичну логіку, тому логічні моделі при всіх своїх перевагах мають істотні обмеження по класах вирішуваних завдань.

Історія штучного інтелекту сповнена драматичних подій, одним з яких став в 1973 р так званий «*доповідь Лайтхілла*», який був підготований в Великобританії на замовлення Британської ради наукових досліджень. Відомий математик *Д. Лайтхілла*, професійно ніяк не пов'язана з іноземними інвестиціями, підготував огляд стану справ в області ШІ. У доповіді були визнані досягнення в галузі ШІ, проте рівень їх визначався як розчарування, і загальна оцінка була негативна з позицій практичної цінності. Цей звіт відкинув європейських дослідників приблизно на років 5 назад, так як фінансування ШІ суттєво скоротилося.

Приблизно в цей же час істотний прорив в практичних додатках штучного інтелекту стався в США, коли до середини 1970-х рр. на зміну пошуків універсального алгоритму мислення прийшла ідея моделювати конкретні знання фахівців-експертів. У США з'явилися перші комерційні *системи, засновані на знаннях*, або *експертні системи (ЕС)*. Прийшов новий підхід до вирішення завдань штучного інтелекту - *подання знань*. Були створені MYCIN і DENDRAL [Shortliffe, 1976; Buchanan, Feigenbaum, 1978], які стали вже класичними, дві перші експертні системи для медицини і хімії. Істотний фінансовий внесок вносить Пентагон, пропонуючи базувати нову програму Міністерства оборони США (Strategic Computer Initiative - SCI) на принципах ШІ. Вже навздогін втрачених можливостей на початку 1980-х рр. проголошена глобальна програма розвитку нових технологій ESPRIT (Європейський Союз), в яку включена проблематика штучного інтелекту.

У відповідь на успіхи США в кінці 1970-х рр. в гонку включається Японія, заявивши про початок проекту машин V покоління, заснованих на знаннях. Проект був розрахований на 10 років і об'єднував кращих молодих фахівців (у віці до 35 років) найбільших японських комп'ютерних корпорацій. Для цих спеціалістів був створений спеціально новий інститут ICOT (Інститут комп'ютерної технології нового покоління). Вони отримали повну «свободу» дій, правда, без права публікації попередніх результатів. В результаті вони створили достатньо громіздкий і дорогий символічний процесор, програмно реалізує ПРОЛОГоподобну мову, яка не отримала широкого визнання. Однак позитивний ефект цього проекту був очевидний. В Японії з'явилася значна група висококваліфікованих спеціалістів у галузі ШІ, яка добилася істотних результатів у різних прикладних задачах. До середини 1990-х рр. японська асоціація ШІ налічує 40 тис. чоловік.

Починаючи з середини 1980-х рр. повсюдно відбувається комерціалізація штучного інтелекту. Ростуть щорічні капіталовкладення, створюються промислові експертні системи, зростає інтерес до самонавчальним системам. Видаються десятки наукових журналів, щорічно збираються міжнародні та національні конференції з різних напрямків ШІ.

ШІ пройшов довгий шлях від моменту свого народження в 1956 р до сьогодення. До ХХІ ст. штучний інтелект залишається однією з найбільш перспективних і престижних областей інформатики (computer science). Головні віхи історії ШІ представлені в «History of Artificial Intelligence» [Stottler, 2002]. Успіхи в наявності, але «блакитна мрія» творців так і не досягнута. Ідеальний інтелектуальний комп'ютер чекає нас попереду.

«Машинний розум» - тест Тьюринга

Алан Тьюринг (Alan M. Turing) вже в 30-і рр. передбачив можливість машинного розуму. Пізніше, в 1950 р Алан Тьюринг в статті «Обчислювальні машини і розум» (*Computing machinery and intelligence*) запропонував тест для перевірки, чи є комп'ютер розумним у людському сенсі слова. Тьюринг запропонував тест, щоб замінити безглузде, на його думку, питання «чи може машина мислити?» на більш чіткіше.

Тест має таку структуру. Суддя (людина) переписується природньою мовою з двома співрозмовниками, один з яких - людина, інший - комп'ютер. Якщо суддя не може надійно визначити, хто є хто, вважається, що комп'ютер пройшов тест. Передбачається, що кожен зі співрозмовників прагне, щоб людиною визнали його. Тест повинен проходити у вигляді переписки повідомленнями. Листування повинна проводитися через контрольовані проміжки часу, щоб суддя не міг робити висновки виходячи з швидкості відповідей. За часів Тьюринга комп'ютери реагували повільніше людини.

Зараз це правило необхідно, тому що комп'ютер реагує набагато швидше, ніж людина. Ідея тесту була нав'язана салонної грою, в ході якої гості намагалися вгадати стать людини, що знаходиться в іншій кімнаті, шляхом написання питань і читання відповідей. В оригінальному формулюванні людина повинна був прикидатися людиною протилежної статі, а тест тривав 5 хвилин.

Тьюринг передбачив, що комп'ютери в кінцевому рахунку пройдуть його тест. Він вважав, що до 2000 р комп'ютер з пам'яттю 1 мільярд біт (близько 119 МБ) в ході 5 хвилинного тесту зможе обдурити суддів в 30% випадків. Цей прогноз не відбувся. Тьюринг також передбачив, що поєднання «мисляча машина» не може вважатися безглуздом, а навчання комп'ютерів буде відігравати важливу роль в створенні потужних комп'ютерів (з чим більшість сучасних дослідників погоджуються). Поки (на 2016 г.) жодна програма не пройшла успішно тест Тьюринга, хоча численні програми можуть створювати ілюзію бесіди.

Першою спробою такого роду була знаменита «Еліза» Вейценбаума. З'являються повідомлення, що деякі програми наблися до 30%. Час розсудить.

В лекціях дається уявлення про те, як представляти в машині досвід і знання професіоналів, але при цьому таємниці здорового глузду, інтуїції і творчості досі не піддаються формалізації. Поки машини «розумні» настільки, наскільки дозволяє їх база знань.

1.2. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ: НАПРЯМКИ РОЗРОБОК

В межах цієї дисципліни в якості робочого визначення можна погодитися з «колективним розумом» Вікіпедії [www.wikipedia.org]:

Штучний інтеле́кт (ШІ, англ. artificial intelligence, AI) — розділ комп'ютерної лінгвістики та інформатики, що опікується формалізацією проблем та завдань, які подібні до дій, що виконує людина. Штучний інтелект — здатність інженерної системи (англ. engineered system) здобувати, обробляти та застосовувати знання та вміння.

При цьому в більшості випадків алгоритм рішення завдання заздалегідь невідомий.

Сьогодні дерево ШІ об'єднує десятки наук, проте максимум зусиль і фінансування мають кілька провідних напрямків, які приведемо надалі.

А) Інтелектуальні ігри та комп'ютерна творчість

За часів ранніх досліджень по ШІ машинний інтелект асоціювався з логічними іграми, і ставилося завдання «навчити» комп'ютер розв'язувати ігрові завдання - шахи, шашки, го. Перші програми на аматорському рівні грали вже в 1957 р. Їх автор, Алекс Бернштейн, також був учасником Дартсмутського семінару. В основі програм лежав один з ранніх підходів - *лабиринтна модель* мислення плюс евристики. Пізніше в 1958 р Ньюелл, Саймон і Шоу вдосконалили алгоритм.

У СРСР цей напрям також було вельми успішним, в зв'язку з традиційною силою вітчизняної шахової та математичної шкіл [Адельсон-Вельський, Арлазаров, Бітман, Донський, Ботвинник]. Так, в 1974 р відбувся міжнародний шаховий турнір машин, забезпечених відповідними програмами. Перемогу на цьому турнірі здобула радянська шахова програма «Каїсса», розроблена за участю М. М. Ботвинника — чемпіона світу з шахів.

Успіх шахової програми Deep Blue фірми IBM, яка виграли навіть у Каспарова, справила величезне враження.

Зараз це швидше комерційний напрямок, так як в науковому плані ці ідеї вважаються тупиковими. Однак, в 2016 р програма компанії Google перемогла чемпіона по грі в шашки го.

Крім того, цей напрямок охоплює твор комп'ютером музики. Основним методом подібні «творчості» є метод *пермутації* (перестановок) плюс використання деяких баз знань і даних, що містять результати досліджень по структурам текстів, рим, сценаріями і т. п. Зараз цей напрямок розглядається вже як «історичне», і всерйоз цим займаються лише окремі групи дослідників.

Б) Подання знань і розробка систем, заснованих на знаннях (Knowledge-based systems)

Саме цим напрямком присвячена ця дисципліна і далі буде детально розглянутий саме цей клас систем III.

В) Машинне навчання (machine learning, data mining, knowledge discovery)

Машинне навчання (machine learning, data mining, knowledge discovery) — область штучного інтелекту, сьогодні активно розвивається і включає моделі, методи і алгоритми, орієнтовані на автоматичне накопичення і формування знань на основі аналізу та узагальнення даних. Використовуються також навчання на прикладах (або індуктивний) і традиційні підходи з теорії розпізнавання образів. Для систем класу machine learning існує «вільний» софт — це системи SLIPPER, Orange, JMLR, Weka3, YALE, Bayda і інші.

В останні роки до цього напрямку тісно примикають системи, що стрімко розвиваються - data mining - аналізу даних і knowledge discovery — пошуку закономірностей.

Г) Розуміння природних мов (natural language processing)

Починаючи з 50-х рр., однією з популярних тем досліджень в галузі ШІ є область комп'ютерної лінгвістики, і зокрема машинного перекладу. Тут треба згадати культову книгу Т.Вінограда «Програма яка розуміє природну мову».

Машинний переклад - виконується на комп'ютері дію з перетворення тексту на одному природною мовою в еквівалентний за змістом текст на іншій мові, а також результат такої дії.

В даний час для розуміння і перекладу використовується кілька більш складних моделей:

- використання так званих «мов-посередників» ;
- *асоціативний пошук* аналогічних фрагментів тексту і їх перекладів в спеціальних текстових репозиторіях або базах даних;
- *структурний підхід* , що включає послідовний аналіз і синтез природно-мовних повідомлень. Традиційно такий підхід має в собі наявність декількох фаз аналізу:
 - *Морфологічний аналіз* - аналіз слів в тексті.
 - *Синтаксичний аналіз* - розбір складу пропозицій і граматичних зв'язків між словами.
 - *Семантичний аналіз* - аналіз змісту складових частин кожної пропозиції на основі деякої предметно-орієнтованої бази знань.

- *Прагматичний аналіз* - аналіз змісту пропозицій в реальному контексті на основі власної бази знань.

Таким чином, система машинного перекладу включає в себе двомовні словники, забезпечені необхідною граматичною інформацією (морфологічної, синтаксичної і семантичної) для передачі перекладних відповідностей.

Приклади показують необхідність участі людини-редактора в перекладі.

Повністю автоматичний переклад в даний час неможливий, а якість програм-перекладачів поки далеко від досконалості (PROMT's Online Translator; Microsoft's Bing Translator; Google Translate; Systran; Babylon; PARS; RUMP та ін.).

Синтез повідомлень на природних мовах (ПМ) дещо легше, ніж аналіз, але також вимагає розгорнутої бази знань.

Серйозний аналіз цієї надзвичайно перспективної області III виходить за межі цієї дисципліни.

Д) Розпізнавання образів (pattern recognition)

Розпізнавання образів (pattern recognition) - один із напрямів ШІ, що виходять із самих його витоків, але в даний час практично виділилися в самостійну науку. Її основний підхід - опис *класів об'єктів* через певні значення значущих ознак.

Кожному об'єкту ставиться у відповідність матриця ознак, по якій відбувається його розпізнавання. Процедура розпізнавання використовує найчастіше спеціальні математичні процедури і функції, що розділяють об'єкти по класах. Цей напрямок близький до машинного навчання, тісно пов'язане з нейрокібернетикою.

Е) Інтелектуальні роботи (robotics)

Ідея створення інтелектуальних роботів далеко не нова. Саме слово «робот» з'явилося в 1920-х рр. як похідне від чеського «робота» - важкої брудної роботи. Його автор - чеський письменник Карел Чапек, який відобразив роботів в своєму оповіданні «РУР».

Роботи - це електротехнічні пристрої, призначені для автоматизації людської праці.

Можна виділити кілька умовних поколінь в історії створення та розвитку робототехніки:

I покоління: *роботи з жорсткою схемою управління* . Практично всі сучасні промислові роботи належать до першого покоління. Фактично це програмовані маніпулятори;

II покоління: *адаптивні роботи з сенсорними пристроями* . Існують зразки таких роботів, але в промисловості вони поки використовуються мало;

III покоління: *самоорганізовані, або інтелектуальні, роботи* . Це є кінцевою метою розвитку робототехніки. Основні невирішені проблеми при створенні інтелектуальних роботів - проблема машинного зору і адекватного зберігання й обробки тривимірної візуальної інформації.

В даний час в світі виготовляється більше 60 000 роботів в рік. фактично робототехніка сьогодні - це інженерна наука, на основі технологій ШІ, але яка ще не готова до їх впровадження в силу різних причин.

У 1980 р Японія досягла лідируючої позиції в світі по виробництву роботів. 50% всіх промислових роботів зроблено і використовується в Японії. З тих пір Японія не втрачає лідируючої позиції, аж до наших днів. У 2000 р компанія Honda розробила робота-гуманоїда Asimo, останні версії якого є одним з найбільш просунутих андроїдів (<http://asimo.honda.com/>). Робот пересувається на досить стійких ногах і має більше ступенів свободи, ніж його попередник P3.

Чемпіонати світу з футболу серед роботів щорічно проводяться з 1997 року та зветься RoboCup. У андроїдів майже все як у справжніх футболістів: голи, нереалізовані моменти, підкати, жовті картки, пенальті.

Основна тенденція в області робототехніки зараз — мініатюризація, аж до нанороботів, що запускаються в кровоносну систему живих організмів.

Однак, в цілому, сучасні роботи продовжують демонструвати механічні успіхи, як і раніше «буксуючи» в області інтелектуальності і проблем машинного зору.

Ж) Нові архітектури комп'ютерів (New hardware platforms and architectures)

Найсучасніші процесори сьогодні засновані на традиційній послідовній архітектурі фон Неймана, використовуваної ще в комп'ютерах перших поколінь. Ця архітектура вкрай неефективна для символної обробки. Тому зусилля багатьох наукових колективів і фірм вже десятки років націлені на розділення апаратних архітектур, спрямованих на обробку символних і логіче-

ських даних. Створюються Пролог- і Лісп-машини, комп'ютери V і VI поколінь. Останні розробки присвячені комп'ютерам баз даних, паралельним і векторним комп'ютерів.

І хоча вдалі промислові рішення існують, висока вартість, недостатнє програмне оснащення і апаратна несумісність з традиційними комп'ютерами істотно гальмують широке використання нових архітектур.

3) Програмне забезпечення систем III

Розробка програмного забезпечення для систем III грає ключову роль. Існують спеціальні мови для вирішення інтелектуальних завдань, в яких акцент ставиться на символічну і логічну обробку, наприклад, LISP, PROLOG, SMALLTALK, РЕФАЛ, STRIPS, POP-11 і ін. Крім цього, створюються пакети прикладних програм, орієнтовані на промислову розробку інтелектуальних систем, або програмні інструментарії штучного інтелекту, наприклад KEE, ARTS, G2, Watson. Досить популярним є також створення так званих порожніх експертних систем або «оболонок»: KRL, OWL, FRL, KAPPA, EXSYS, E2glite і ін., які можна наповнювати базами знань, створюючи різноманітні експертні системи.

I) Багатоагентні системи

В одному з найбільш популярних сучасних підручників з ШІ Рассела і Норвіга під агентом розуміється «будь-яка сутність, яка знаходиться в деякому середовищі, сприймає її за допомогою сенсорів, отримуючи дані, які відображають події, що відбуваються в середовищі, інтерпретує ці дані і діє на оточуюче середовище за допомогою ефекторів. Таким чином, тут вирізняються чотири вихідних агентоутворюючих чинника - середовище, сприйняття, інтерпретація, дія.

Розробка технології *штучних агентів*, створення *мультиагентних систем* (МАС) є однією з найбільш важливих і перспективних областей розвитку ШІ. У фахівців існує уявлення про *штучних агентів* як *активних, автономних, комунікабельних, а головне - мотивованих об'єктах*, які «живуть» і «діють» в *складних, динамічних і найчастіше у віртуальних середовищах*.

Агентно-орієнтований підхід знаходить застосування в таких областях, як розподілене рішення складних завдань, поєднане проектування виробів, реінжиніринг бізнесу і побудова віртуальних підприємств, імітаційне моделювання інтегрованих виробничих систем і електронна торгівля, організація роботи колективів роботів і розподілена (поєднана) розробка комп'ютерних програм. У найближчому майбутньому він, можливо, займе центральне місце при розвитку засобів управління інформацією і знаннями, при створенні та впровадженні новітніх систем телекомунікації, розвитку глобальних комп'ютерних мереж.

При визначенні поняття «агент» часто спираються на уявлення про об'єкт, розвинений школою об'єктно-орієнтованого програмування (ООП).

Тоді *штучний агент* можна розглядати як *метаоб'єкт* , наділений деякою часткою *суб'єктності* , тобто здатністю маніпулювати іншими об'єктами, створювати та знищувати їх, а також має розвинуті засоби взаємодії з середовищем і собі подібними. Набір базових характеристик агента включає:

- 1) *активність* - здатність до організації і реалізації дій;
- 2) *автономність* (напівавтономність) - відносна незалежність від довкілля або наявність деякої «свободи волі», пов'язаної з гарним ресурсним забезпеченням його поведінки;
- 3) *товариськість* - впливає з необхідності вирішувати свої завдання разом з іншими агентами та забезпечується розвиненими протоколами комунікації;
- 4) *цілеспрямованість* - припускає наявність власних джерел *мотивації* , а в більш широкому плані - спеціальних інтенціональних характеристик.

Таке уявлення близько до одного з найбільш популярних визначень поняття «агент» у М. Вулдрідж і Н. Дженнінгса.

Іноді агентні технології називають розподіленим ШІ (рші) (Distributed AI). Ядро РШІ становлять дослідження взаємодії невеликої кількості інтелектуальних агентів, які включають бази знань і вирішувачі. Головною проблемою в РШІ є розробка інтелектуальних груп і організацій, здатних вирішувати завдання шляхом міркувань, пов'язаних з обробкою символів. Це передбачає узгодження цілей, інтересів і стратегій різних агентів, координацію дій, вирішення конфліктів шляхом переговорів; теоретичну базу тут складають результати, отримані в психології малих груп та соціології організацій.

Важливим розділом РШ є кооперативне розподілене рішення задач (КРРЗ). Йдеться про мережу слабо пов'язаних між собою розв'язувачів, які спільно працюють з метою вирішення завдань, що виходять за рамки індивідуальних можливостей. Різні вузли подібної мережі, як правило, мають неоднаковий досвід (знання, точки зору) і різні ресурси.

Досить активно в розробці МАС проявляється напрям, під назвою «штучне життя» (ШЖ) (у вузькому сенсі терміна artificial life). Воно пов'язане з трактуванням інтелектуальної поведінки в контексті виживання, адаптації та самоорганізації в динамічному середовищі. У руслі ШЖ глобальна поведінка всієї системи розглядається як результат локальних взаємодій простих і необов'язково інтелектуальних агентів. Тут також використовуються терміни «колективний інтелект» (Collective intelligence), або «інтелект рою» (swarm intelligence). Типовими прикладами такого колективного інтелекту з біології є колонії мурах, бджолині вулики і т. п.

Відповідно, тут базовими дисциплінами можуть служити різні галузі біологічної науки і в першу чергу - еволюційної теорії та генетики.

Існують спеціальні програмні середовища і мови для взаємодії агентів, наприклад, KIF, KQML, FIPA-ACL і ін. Останнім часом активно використовується онтологічний підхід.

Інші напрямки ШІ

Як міждисциплінарна наука, ШІ постачає живлення від суміжних дисциплін - психології, математики, лінгвістики, філософії, нейрофізіології і декількох інших. Вище перераховані лише ті напрямки, які прямо або побічно пов'язані з основною тематикою даної дисципліни - інженерією знань. Варто лише поглянути на основні рубрикатори конференцій зі ШІ, щоб зрозуміти, наскільки широко простягається область цих досліджень:

- автоматичне доведення теорем (theorem proving);
- автоматичні міркування (automated reasoning);
- автоматичне навчання (machine learning);
- агенти і багатоагентні системи (agents and multi-agent systems);
- Байєсові мережі (Bayesian networks);
- віртуальна реальність (virtual reality);
- генетичні алгоритми (evolutionary computation, genetic algorithms);
- дедуктивні моделі (deductive models);

- гри (game playing);
- інженерія знань (knowledge engineering);
- інтелектуальні бази даних (intelligent databases);
- інженерія знань (knowledge engineering);
- інтелектуальні інтерфейси (intelligent interfaces);
- інтелектуальний Інтернет (інтелектуальна павутина) (Internet / WWW intelligence, Semantic Web);
- штучне життя (artificial life);
- когнітивне моделювання (cognitive modeling);
- концептуальні карти-графи (concept maps and graphs);
- логіки (індуктивні, дескриптивні, нечіткі та ін.) (Logics (inductive / descriptive / fuzzy, etc.);
- логіки простору і часу (spatial / temporal reasoning);
- машинна творчість (music, Art and AI);
- машинне зір (vision);

- машинний переклад (machine translation);
- менеджмент знань (knowledge management);
- м'які обчислення (soft computing);
- нейронні мережі (neural networks);
- онтології (ontologies);
- роботи (robotics);
- вилучення знань (data mining, knowledge discovery);
- розпізнавання та синтез мови (speech processing);
- розподілений ІІІ (distributed AI);
- розв'язання задач (problem solving);
- розуміння природної мови (natural language processing);
- логічні висновки (computer logics);
- міркування на прецедентах (case-based reasoning);
- подання знань (knowledge representation);
- придбання знань (knowledge acquisition);

- формальні моделі (formal models);
- людино-машинні системи (human-computer interaction);
- евристики (heuristics); і багато іншого.

Далі в рамках наступних лекцій докладно буде розглянуто напрямки інженерії знань.

1.3. СИСТЕМИ, ЗАСНОВАНІ НА ЗНАННЯ, ТА ІНЖЕНЕРІЯ ЗНАНЬ

Подання знань і розробка систем, заснованих на знаннях (knowledge-based systems), - це одне з основних напрямків штучного інтелекта. Воно пов'язане з розробкою моделей представлення знань, створенням баз знань, які становлять ядро експертних систем. Останнім часом включає в себе моделі та методи добування та структурування знань і зливається з інженерією знань. Саме дослідженням в цій області присвячена дана книга. Центральна парадигма інтелектуальних технологій сьогодні - це обробка знань. Системи, ядром яких є база знань або модель предметної області, описана на мові надвисокого рівня, наближеному до природного (МПЗ - мові представлення знань), називають *інтелектуальними*. Найчастіше інтелектуальні системи (ІС) застосовуються для розв'язання складних завдань, де основна складність вирішення пов'язана з використанням слабоформалізованих знань фахівців-практиків і де логічна (або смислова) обробка інформації превалює над обчислювальною. Наприклад, підтримка

прийняття рішення в складних ситуаціях, постановка діагнозу і рекомендації щодо методів лікування, аналіз візуальної інформації, управління диспетчерськими пультами й таке ін.

Спрямованість цієї дисципліни дозволить студентіві далі докладно познайомитися з основними технологіями розробки систем, заснованих на знаннях, - від домашинного етапу придбання знань до програмних систем реалізації.

Практично у всіх областях інформатики в даний час відслідковується тенденція випередження технологічних засобів розробки по відношенню до їх теоретичного обґрунтування. Таке ж становище існує в області інтелектуальних систем, заснованих на парадигмі обробки знань (експертні системи, лінгвістичні процесори, навчальні системи і т. ін.).

З одного боку, це пояснюється тим, що з перших кроків наука про ШІ була спрямована на моделювання слабоформалізованих смислових завдань, в яких не застосовується традиційний математичний апарат; з іншого боку, ШІ - це гілка інформатики і активно розвивається як промислова індустрія програмних засобів в умовах жорсткої конкуренції, де часом важливіше швидке впровадження нових ідей і підходів, ніж їх аналіз і теоретична проробка.

Необхідність розробки теоретичних основ науки про методи розробки систем, заснованих на знаннях - інженерії знань, - були обґрунтовані в низці робіт Р. Шенка, М. Мінського, Д. О. Поспелова, Є. В. Попова - провідних фахівців в області ШІ. Перші кроки

Так і з'явилася інженерія знань, наука, присвячена теоретичним і практичним проблемам проектування баз знань - отримання і структуруванні знань фахівців для подальшої розробки інтелектуальних систем або систем управління знаннями.

1.4. ПРО ЕКСПЕРТНІ СИСТЕМИ

Експертні системи (ЕС) - це найбільш поширений клас ІС, орієнтований на тиражування досвіду висококваліфікованих спеціалістів в галузях, де якість прийняття рішень традиційно залежить від рівня експертизи, наприклад, медицина, юриспруденція, геологія, економіка, військова справа та ін.

Сучасні ЕС - це складні програмні комплекси, що акумулюють знання фахівців в конкретних предметних областях і поширюють цей емпіричний досвід для консультацій менш кваліфікованих користувачів. Розробка експертних систем активно розвивається як галузь інформатики та спрямована на використання комп'ютерів для обробки інформації в тих областях науки і техніки, де малоприматні традиційні математичні методи моделювання. У цих областях важливі смислова та логічна обробка інформації, досвід експертів.

Фактично інтелектуальні прикладні системи використовуються зараз в тисячах комп'ютерних додатків. Ринок інтелектуальних систем в 80-90-і рр. ріс лавиноподібно. Надалі майже 30% -вий приріст доходу змінився більш плавним зростанням темпів. Зараз річний обсяг світового ринку інтелектуальних систем International Data Corporation (IDC) оцінила в \$ 1,4 млрд (на 2014 г.). При цьому приблизно 20% витрат припадає на Японію, близько 25% - на США і Канаду і більше 40% - на європейські країни. А річний дохід від продажів штучних нейронних мереж становить \$ 6,3 млрд (за даними компанії Computer Associates).

ЕС ефективні лише в специфічних «експертних» областях, де важливий емпіричний досвід фахівців. Наприклад, компанія American Express скоротила свої втрати на 27 млн дол. в рік завдяки експертній системі, яка визначає доцільність видачі або відмови в кредиті тій чи іншій фірмі.

Щорічно великі фірми розробляють десятки ЕС типу «in-house» для внутрішнього користування. Ці системи інтегрують досвід фахівців компаній за ключовими і стратегічно важливими технологіями.

Перші системи такого роду MYCIN і DENDRAL з'явилися в США в середині 1970-х рр. В даний час в світі нараховується кілька тисяч промислових ЕС, які дають поради в різних випадках, наприклад: аналіз ДНК - FSSI; управління складними диспетчерськими пультами, наприклад, мережі розподілу електроенергії - Alarm Analyser; прийняття особистих рішень - індивідуальні мобільні порадики; допомога при шопінгу - DIY-Expertbots; постановки медичних діагнозів - PXDES (аналіз рентгенограм), ARAMIS, NEUREX; відповіді на муніципальні запити від жителів - PublicAccess Expert System; пошук несправностей в електронних приладах, діагностика відмов контрольно-вимірювального обладнання - Intelligence Ware, Plant Diagnostics, FOREST; прогноз військових дій - ANALYST, BATTLE; проектування інтегральних мікросхем - DAA, NASL, QO; формування портфеля інвестицій, оцінка фінансових ризиків - RAD; оподаткування та заповнення податкових декларацій - Taxadvisor, Investor and Financial Advisor, ExperTax, RUNE; управління компанією - PortBLue; управління перевезеннями - AIRPLAN і т. д.

Чинники, що свідчать про необхідність розробки і впровадження експертної систем: структурувати знання, визначати і формалізувати поняття і правила, необхідні для зання проблеми.

Основні складові ЕС:

- *Інтерфейс користувача* - комплекс програм, що реалізують діалог користувача з ЕС як на стадії введення інформації, так і при отриманні результатів.
- *База знань (БЗ)* - ядро ЕС, сукупність знань предметної області, записана на машинний носій в формі, зрозумілою експерту і користувачу (зазвичай на деякій мові, наближеній до природної). паралельно такому «людському» поданню існує БЗ у внутрішньому «машинному» поданні.
- *Вирішувач* - програма, що моделює хід міркувань експерта на основі знань, наявних в БЗ. Синоніми: *дедуктивна машина, машина виведення, блок логічного висновку.*

- *Підсистема пояснень* - програма, що дозволяє користувачеві знайти відповіді на питання: «Як була отримана та чи інша рекомендація?» і «Чому система прийняла таке рішення?» Відповідь на питання «як» - це трасування процесу отримання рішення із зазначенням використаних фрагментів БЗ, тобто основних кроків ланцюга розміркувань. Відповідь на запитання «Чому» - посилення на умовивід, безпосередньо передувала отриманого рішенням, тобто відхід на один крок назад. Розвинені підсистеми

пояснень підтримують і інші типи питань.

- *Інтелектуальний редактор БЗ* - програма, що представляє інженеру за знаннями можливість створювати БЗ в діалоговому режимі. Включає в себе систему вкладених меню, шаблонів мови представлення знань, підказок («Help» - режим) і інших сервісних засобів, що полегшують роботу з базою.

Клас ЕС сьогодні об'єднує кілька тисяч різних програмних комплексів, які можна класифікувати за різними критеріями. корисними можуть виявитися класифікації, представлені на малюнку 1.



Мал. 1

Класифікація експертних систем

Класифікація за розв'язуваною задачею

Розглянемо зазначені на малюнку типи завдань докладніше.

Інтерпретація даних . Під інтерпретацією розуміється процес визначення сенсу даних, результати якого повинні бути узгодженими і коректними.

Приклади:

- інтерпретації тестів на наявність алергії, заснована на технології мікрочіпів, Allergenius;
- кредитний скоринг HIS (Hybrid credit ranking intelligent system) - швидка, точна і стійка процедура оцінки кредитного ризику, що має наукове обґрунтування;
- визначення основних властивостей особистості за результатами психодіагностичного тестування в системах АВТАНТЕСТ і МІКРОЛЮШЕР і ін.

Діагностика . Під діагностикою розуміється процес співвідношення об'єкту з деяким класом об'єктів і / або виявлення несправності в деякій системі. Несправність - це відхилення від норми. Таке трактування дозволяє з єдиних теоретичних позицій розглядати і несправність обладнання в технічних системах, і захворювання живих організмів, і всілякі природні аномалії. Приклади:

- діагностика і терапія звуження коронарних судин - ANGY;
- діагностика захворювань рису - MANAGE;
- діагностика, класифікація і корекція терапії гострих розладів кровообігу у дітей - «Айболить».

Моніторинг . Основне завдання моніторингу - безперервна інтерпретація даних в реальному масштабі часу і сигналізація про вихід тих або інших параметрів за допустимі межі. Головні проблеми - «пропуск» тривожної ситуації і інверсне завдання «помилкового» спрацьовування. Приклади:

- контроль за роботою електростанцій - СПРiНТ;
- допомогу диспетчерам атомного реактора - REACTOR;
- контроль аварійних датчиків на хімічному заводі - FALCON;
- оцінка дій космонавта в процесі тренування щодо ручного стикування транспортного вантажного корабля (ТКГ) і Міжнародної космічної станції (МКС);
- допомогу інвесторам у виборі акцій для складання інвестиційного портфоліо - PORSEL (PORtfolio SElection system) і ін.

Проектування . Проектування полягає в підготовці специфікацій на забудівлю «об'єктів» із заздалегідь визначеними властивостями. Під специфікацією розуміється весь набір необхідних документів - креслення, пояснювальна записка

і т. д. Основні проблеми тут - отримання чіткого структурного опису знань про об'єкт і проблему «сліду». Приклади:

- проектування конфігурацій EOM VAX - 11/780 в системі XCON (або R1);
- проектування великих інтегральних схем CADHELP;
- синтез електричних ланцюгів - SYN;
- система управління контролю доступу, концептуальна модель, дозволяюча проектувати сценарії медичних запитів доступу до даних - SitBAC (Situation-Based Access Control) і ін.

Прогнозування . Прогнозування дозволяє передбачати наслідки деяких подій або явищ на підставі аналізу наявних даних. Прогнозуючі системи логічно виводять вірогідні наслідки з заданих ситуацій. Приклади:

- передбачення погоди - система WILLARD;
- оцінки майбутнього врожаю - PLANT;
- прогнози в економіці - ECON і ін.

Планування . Під плануванням розуміється пошук планів дій, відносяться до об'єктів, здатним виконувати деякі функції. У таких ЕС використовуються моделі поведінки реальних об'єктів з тим, щоб логічно вивести наслідки планованої діяльності. Приклади:

- планування поведінки робота - STRIPS;
- планування промислових замовлень - ISIS;
- планування туристичної діяльності - SigTur / E-Destination;
- планування експериментів з клонування генів у молекулярній генетиці - MOLGEN.

Навчання . Під навчанням розуміється використання комп'ютера для навчання якійсь дисципліни або предмету. Системи навчання діагностують помилки при вивченні будь-якої дисципліни за допомогою комп'ютера і підказують правильність рішення. Вони акумулюють знання про гіпотетичного «учня» і його характерні помилки, здатні діагностувати слабкості в пізнаннях учнів і знаходити відповідні засоби для їх ліквідації.
приклад:

- вивчення мови програмування ЛІСП в системі «Вчитель ЛІСП»;
- система PROUST - вивчення мови Паскаль;
- середовище розробки широкого класу навчальних ЕС - АТ-ТЕХНОЛОГІЯ і ін.

Управління. Під управлінням розуміється функція організованої системи, що підтримує певний режим діяльності. Такого роду ЕС здійснюють управління поведінкою складних систем відповідно до заданих специфікаціями. Приклади:

- допомога в управлінні газової котельні - GAS;
- управління сільськогосподарським комплексом - CALEX;
- управління системою календарного планування Project Assistant;
- управління процесом очищення стічних вод (ОСВ) від нафтового забруднення на біологічних очисних спорудах.

Підтримка прийняття рішень. Підтримка прийняття рішення - це сукупність процедур, що забезпечує особа, яка приймає рішення, необхідної інформацією і рекомендаціями для полегшення процесу прийняття рішення. Приклади:

- вибір стратегії виходу фірми з кризової ситуації - CRYISIS;
- поради по боротьбі з шкідниками врожаю - LIMEX;
- підтримка прийняття рішень для оптимізації роботи з валютними опціонами - Nereid;
- допомога у виборі страхової компанії або інвестора - CHOICE;
- поради зі стратегічного управління і планування діяльності підприємства трубопровідного транспорту нафтопродуктів.

Класифікація за зв'язком з реальним часом

Статичні ЕС розробляються в предметних областях, в яких база знань і інтерпретовані дані не змінюються в часі. Вони стабільні.

Приклад: діагностика несправностей в автомобілів.

Квазідинамічні ЕС інтерпретують ситуацію, яка змінюється з деяким фіксованим інтервалом часу.

Приклад: мікробіологічні ЕС, в яких знімаються лабораторні вимірювання з технологічного процесу один раз в 4-5 годин (виробництво лізину, наприклад) і аналізується динаміка отриманих показників по відношенню до попереднього виміру.

Динамічні ЕС працюють в сполученні з датчиками об'єктів у режимі реального часу з безперервною інтерпретацією надходять в систему даних.

Приклад:

- управління гнучкими виробничими комплексами, моніторинг в реанімаційних палатах і т. д.
- програмний інструментарій для розробки динамічних систем G2.

Класифікація за ступенем інтеграції з іншими програмами

Автономні ЕС працюють безпосередньо в режимі консультацій з користувачем для специфічних «експертних» завдань, при вирішенні яких не вимагається залучати традиційні методи обробки даних (розрахунки, моделювання і т. д.).

Гібридні ЕС представляють програмний комплекс, котрий агрегує стандартні пакети прикладних програм (ППП) (наприклад, математичну статистику, лінійного програмування або системи управління базами даних) і засоби маніпулювання знаннями. Це може бути інтелектуальна надбудова над PPP або інтегроване середовище для вирішення складного завдання з елементами експертних знань.

Про колектив розробників експертної системи

Над створенням ЕС працює *колектив розробників* (КР), під яким будемо розуміти всю групу фахівців, відповідальних за створення ЕС. До складу КР входять принаймні три людини - користувач, експерт й інженер по знаннях. Зазвичай в КР включають і програміста. Таким чином, мінімальний склад КР включає чотири людини; реально ж він розростається до 8-10 чоловік. Часто в цей колектив входять менеджер і його технічний помічник.

Якщо використовувати аналогії з близьких галузей, то КР найбільше схожий з групою адміністратора бази даних при побудові інтегрованих інформаційних систем або бригадою програмістів, що розробляють складний програмний комплекс. При відсутності професійного менеджера керівником КР, який беруть участь у всіх стадіях розробки, є аналітик або інженер по знаннях, тому до його кваліфікації пред'являються високі вимоги.

Для забезпечення ефективності співпраці будь-якої творчої групи, необхідно виникнення атмосфери взаєморозуміння і довіри, яке, в свою чергу, обумовлено психологічною сумісністю членів групи; отже, при формуванні КР повинні враховуватися психологічні властивості учасників. Нижче наведені два аспекти характеристик членів КР: А - психофізіологічний, Б - професійний.

Користувач

А. К користувачеві пред'являються найслабші вимоги, оскільки його не вибирають. Бажані якості: а) дружелюбність; б) вміння пояснювати свої вимоги до системи; в) відсутність психологічного бар'єру до застосування обчислювальної техніки; г) інтерес до нового. Від користувача залежить, чи буде застосовуватися розроблена ЕС. Помічено, що найбільш яскраво якості в) і г) виявляються в молодому віці, тому іноді такі користувачі охочіше застосовують ЕС, при цьому у них відсутній комплекс неповноцінності через те, що комп'ютер їм щось підказує.

Б. Необхідно, щоб користувач мав деякий базовий рівень кваліфікації, який дозволить йому правильно зрозуміти рекомендації ЕС.

Експерт

А. Експерт в кінцевому рахунку задає рівень компетенції бази знань, є важливою фігурою в групі КР. Бажані якості:

а) доброзичливість; б) готовність поділитися своїм досвідом; в) вміння пояснити (педагогічні навички); г) інтерес (моральний, а краще ще й матеріальний) до успішності розробки. Вік експерта може виявитися зрілим, що необхідно враховувати всім членам групи.

Б. Крім безумовно високого професіоналізму в обраній предметній області, бажано знайомство експерта з популярною літературою зі штучного інтелекту і експертних систем для більш ефективного процесу здобування знань.

Програміст

А. Відомо, що програмісти мають найнижчу потребою в спілкуванні серед представників різних професій. Однак при розробці ЕС необхідний тісний контакт членів групи, тому бажані такі якості:

а) товариськість; б) здатність відмовитися від традиційних навичок та освоїти нові методи; в) інтерес до розробки.

Б. Оскільки сучасні ЕС - найскладніші і дорогі програмні комплекси, програмісти в КР повинні мати досвід і навички розробки програм. Обов'язково знайомство з основними структурами представлення знань і механізмами виведення, станом вітчизняного і світового ринку програмних продуктів для розробки ЕС і діалогових інтерфейсів.

Інженер по знаннях

Зростає фактичний попит на унікальних фахівців по роботі зі знаннями - аналітиків. На сучасному ринку високих інформаційних технологій потреба в аналітиках багаторазово перевищує попит на програмістів.

Цих фахівців називають по-різному - системні аналітики, постановники завдань, інженери по знаннях, інженери-когнітологи (анг. - knowledge engineers). Близькі поняття і синоніми: постановник завдань / СКО (Chief knowledge officer) - директор з управління знаннями / менеджер знань / функціональний аналітик або бізнес-аналітик.

А. Існують такі професії і види діяльності, для яких природні якості особистості (спрямованість, здібності, темперамент) можуть мати характер абсолютного показання або протипоказання до занять. Мабуть інженерія знань належить до таких професій. За різними оцінками, це одна з найбільш малочисельних, високооплачуваних і дефіцитних в світі спеціальностей. Спробуємо дати начерки до портрета інженера по знанням (без претензії на повноту і точність визначень).

Стать . Психологи стверджують, що чоловіки більш схильні до широкого охоплення явищ і в середньому у них вище аналітичність, надзвичайно корисна інженеру по знаннях, якому треба мати розвинуте логічне мислення та вміння оперувати складними формальними структурами. З іншого боку, відомо, що у жінок вище спостережливість до окремих деталей об'єктів. Так що стать не є визначаючим показником або протипоказанням до цієї професії.

Стиль спілкування. Інженер по знаннях “задає тон” у спілкуванні з експертом, він веде діалог, і від нього у кінцевому рахунку залежить продуктивність. Можна розрізнити два стилі спілкування: діловий або жорсткий та дружній або м’який, делікатний. Дружній буде свідомо більш успішним, оскільки знижує “ефект фасаду” у експерта, розслаблює його. Делікатність, уважність, інтелігентність, ненав’язливість, скромність, вміння слухати та задавати питання, гарна комунікабельність і в той же час впевненість у собі — ось той стиль, який рекомендується.

Портрет інженера по знаннях можна доповнити іншими характеристиками — широтою поглядів і інтересів, артистичністю, відчуттям гумору, шармом і таким іншим.

Б. При визначенні професійних вимог до аналітика, варто враховувати, що йому необхідні різноманітні навички та вміння для грамотного і ефективного проведення процесів отримання, концептуалізації і формалізації знань.

Успішність вибору і підготовки колективу розробників ЕС визначає ефективність і тривалість всього процесу розробки.

1.5. Розробка інтелектуальних систем

За розробку експертних систем має сенс братися компанії, в якій накопичуваний досвід автоматизації рутинних процедур обробки інформації, таких як, наприклад, формування корпоративних інформаційних систем або організація складних розрахунків. В процесі розв'язання таких задач, по перше, відбувається підготовка аналітиків, по друге, з'являється можливість виокремити від експертних систем неекспертні задачі.



Рис.2 Етапи розробки експертної системи

Процес розробки промислової експертної системи, спираючись на традиційні технології, можна розбити на шість більш менш незалежних етапів, практично незалежних від предметної області (рис.2).

Послідовність етапів дана тільки з метою отримання загального представлення про процес створення ідеального проекту. Звичайно, послідовність ця не цілком фіксована. Насправді кожен наступний етап розробки здатний принести нові ідеї, які можуть вплинути на попередні рішення і навіть привести до їх перегляду і переробки. Саме тому багато фахівців з інформатики дуже критично ставляться до методології експертних систем. Вони вважають, що на розробку таких систем витрати і занадто великі, а отримані в результаті програми вимагають значних обчислювальних ресурсів.

Правильний вибір проблеми представляє критичну частину розробки в цілому. Якщо вибрати невідповідну проблему, можна загрузнути в «болоті» задач, які ніхто не знає, як вирішувати. Невідповідна проблема може також привести до розробки експертної системи, створення якої буде коштувати більше, ніж передбачувана економія від її впровадження. Цей етап традиційно включає:

- визначення проблемної області та завдання;
- знаходження експерта, який бажає співпрацювати при вирішенні проблеми, і призначення колективу розробників;
- підготовку докладного плану розробки.

Експертна система не замінює «класичну» автоматизацію (бази даних, статистику, документообіг та інше), а доповнюють її.

Експертні системи розробляються шляхом отримання специфічних знань від експерта та введення їх в систему. Деякі системи можуть містити стратегії одного індивіда, отже, знайти підходящого експерта - це ключовий крок у створенні експертних систем.

Перш за все, інженер по знаннях повинен переконатися, що на перераховані нижче питання буде отримано позитивну відповідь:

- дана задача може бути вирішена за допомогою експертної системи;
- експертну систему можна створити пропонованими на ринку засобами;
- є відповідний експерт;
- запропоновані критерії продуктивності є розумними;
- витрати і термін їх окупності прийнятні для замовника.

Спочатку розробляється прототип (усічена ЕС), при незадовільному функціонуванні прототипу експерт і інженер по знаннях мають можливість оцінити, що саме буде включено в розробку остаточного варіанту системи.

Якщо спочатку вибрані об'єкти або властивості виявляються непригодними, їх слід змінити. Можна зробити оцінку загального числа евристичних правил, необхідних для створення остаточного варіанта експертної системи. Іноді [Хювянен, Сеппянен, 1991] при розробці промислової та / або комерційної системи виділяють додаткові етапи для переходу: *демонстраційний прототип - діючий прототип - промислова система - комерційна система.*

Однак частіше реалізується плавний перехід від демонстраційного прототипу до промислової системи, при цьому, якщо програмний інструментарій був обраний вдало, не обов'язково навіть переписувати остаточний варіант іншими програмними засобами.

Поняття ж комерційної системи у нас входить в поняття промисловий програмний продукт, або промислової ЕС в цій дисципліні.

Основна робота при доопрацюванні системи полягає в істотному розширенні бази знань, тобто додаванні великої кількості додаткових правил, фреймів, вузлів семантичної мережі або інших елементів знань. Після встановлення основної структури знань ЕС інженер по знаннях приступає до розробки та адаптації інтерфейсів, за допомогою яких система буде спілкуватися з користувачем і експертом.

На цьому етапі розробки більшість експертів дізнаються доволі про введення правил і можуть самі вводити в систему нові правила. Таким чином, починається процес, під час якого інженер по знаннях передає право власності і контролю за системою експерту для уточнення, детальної розробки і обслуговування.

Після завершення етапу розробки промислової експертної системи необхідно провести її тестування щодо критеріїв ефективності.

Оцінку можна проводити виходячи з різних критеріїв, згрупованих таким чином:

- критерії користувачів (зрозумілість і «прозорість» роботи системи, зручність інтерфейсів і ін.);
- критерії запрошених експертів (оцінка порад-рішень, запропанованих системою, порівняння її з власними рішеннями, оцінка підсистеми пояснень та ін.);
- критерії колективу розробників (ефективність реалізації, продуктивність, час відгуку, дизайн, широта охоплення предметної області, несуперечливість БЗ, кількість тупикових ситуацій, коли система не може прийняти рішення, аналіз чутливості програми до незначних змін в уявленні знань, вагових коефіцієнтах, застосовуваних у механізмах логічного висновку, даних і т. п.).

Далі здійснюється стикування експертної системи з іншими програмними коштами в середовищі, в якій вона буде працювати, і навчання людей, яких вона буде обслуговувати.

Технологія швидкого прототипування

Найважливішу роль при розробці ЕС грає технологія прототипування.

Прототипна система є усіченою версією експертної системи, зпроектованою для перевірки правильності кодування фактів, зв'язків і стратегій міркувань експерта.

Обсяг прототипу - кілька десятків правил, фреймів або прикладів. На рисунку 3 зображені шість стадій розробки прототипу і мінімальний колектив розробників, зайнятих на кожній стадії . Наведемо коротку характеристику кожної з стадій, хоча ця схема являє грубе наближення до складного ітеративному процесу побудови ЕС.

Хоча будь-який теоретичне поділ буває часто умовним, усвідомлення колективів розробників чітких завдань кожної стадії представляється доцільним. Терміни наведені умовно, тому що залежать від кваліфікації спеціалістів і особливостей завдання.

Ідентифікація проблеми . На цьому етапі відбувається знайомство і навчання членів колективу розробників, а також створення неформального визначення проблеми. Середня тривалість 1-2 тижні. Уточнюється завдання, планується хід розробки прототипу експертної системи. Також визначаються:

- необхідні ресурси (час, люди, комп'ютери і т. ін.);
- джерела знань (книги, додаткові експерти, методики);
- наявні аналогічні експертні системи;
- цілі (поширення досвіду, автоматизація рутинних дій і ін.);
- класи розв'язуваних завдань і т. ін.

Вилучення знань включає отримання інженером по знаннях найбільш повного з можливих уявлення про предметну область і способи прийняття рішення в ній. Середня тривалість - 1-3 місяці. Відбувається перенесення компетентності експертів на інженерів по знаннях з використанням різних методів.

Структурування, або концептуалізація, знань. Проводиться розробка неформального опису знань про предметну область у вигляді графа, таблиці, діаграми або тексту, яке відображує основні концепції і взаємодії зв'язку між поняттями предметної області. Такий опис називається *полем знань*. Середня тривалість етапу - 2-4 тижні. Детально стадія структурування буде подана далі.

Формалізація, або кодифікація, знань. Будується формалізоване подання концепцій предметної області на основі обраної *мови представлення знань* (МПЗ). Виконується розробка бази знань на мові представлення знань, яка, з одного боку, відповідає структурі поля знань, а, з іншого, дозволяє реалізувати прототип системи на наступній стадії програмних реалізацій. Середня тривалість - 1-2 місяці. Детально мови представлення знань будуть розглянуті надалі.

Реалізація. Попередній підхід до програмної реалізації задачі визначається виходячи з характеристик завдання і ресурсів, виділених на її рішення.

Середня тривалість - 1-2 місяці. Найчастіше перший прототип відкидається на етапі реалізації діючої ЕС. Створюється прототип експертної системи, що включає базу знань і інші блоки, за допомогою одного з таких способів:

- програмування на традиційних мовах типу PYTHON, C #, Java та ін .;
- програмування на спеціалізованих мовах, застосовуваних в задачах штучного інтелекту: LISP, ПРОЛОГ, SMALLTALK та ін .;
- програмування на спеціалізованих мовах представлення знань KRL, OWL, FRL та ін.;
- використання інструментальних засобів розробки ЕС типу CLIPS, ПІЕС, Simer + MIR, G2 ;
- використання «порожніх» ЕС або «оболонки» типу EXSYS, KAPPA та ін.

Середня тривалість - 1-2 місяці. Більш детально ці питання розрозглядаються далі.

Перед розробниками ЕС часто гостро стоїть питання про вибір адекватного програмного інструментарію розробки. Швидко змінюється ринок засобів розробки пропонує десятки програмних систем, однак при виборі треба брати до уваги цілий ряд як об'єктивних, так і суб'єктивних чинників. Єдиним непорушним правилом є наступне:

«Вибір системи реалізації повинен здійснюватися ПІСЛЯ етапу надбання і структурування знань».

Значущим чинником при виборі засобу розробки є структура знань предметної області, заздалегідь передбачити яку часто складно. Основні фактори, що впливають на вибір програмного інструменту:

- особливості предметної області та наявність кваліфікованої команди розробників;
- бюджет проекту і необхідність стикування з іншими програмними продуктами;
- кількість користувачів майбутньої системи і їх кваліфікація.

Наведемо деякі приклади інструментів розробки, згідно наведених класифікації.

Мови програмування. Основні мови реалізації були перераховані вище. Але фаворитом в Європі і Японії залишається PROLOG, а в США - LISP.

Оболонки. Особливою популярністю користуються «порожні» ЕС або оболонки через простоту і швидкості розробки БЗ.

Програмні середовища або інструментарії . Програмні інструментальні системи (наприклад, CLIPS (C Language Integrated Production System)) включають в себе спеціальні мови (наприклад, об'єктно-орієнтована мова COOL) для написання експертних систем. Далі за допомогою цієї мови описується база знань.

Окремо слід згадати ЕС реального часу, для розробки яких використовуються більш складні і дорогі інструменти. Наприклад, G2 - програмний продукт, призначений для швидкої розробки експертних систем реального часу, які відстежують і виявляють неполадки в складних програмно-апаратних комплексах. Перша версія інструментальної системи G2 була розроблена в 1988 р компанією Gensym. Це потужне інструментальний засіб , що володіє всіма властивостями, необхідними для створення експертних систем реального часу.

Тестування. Етап включає виявлення помилок в підході і реалізації прототипу та вироблення рекомендацій з доведення системи до промислового варіанту. Оцінюється і перевіряється робота прототипу з метою приведення у відповідності до реальних запитів користувачів. Середня тривалість - 1-2 тижні. Перевіряються в основному такі властивості прототипу:

- зручність і адекватність інтерфейсів вводу-виводу (характер питань в діалозі, зв'язність виведеного тексту результату і ін.);
- ефективність стратегії управління (порядок перебору, використання нечіткого виведення і ін.);
- коректність бази знань (повнота і несуперечність правил).

Середня тривалість - 1-2 тижні.

1.6. ПРИКЛАДНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Класичним прикладом експертної системи є система FocalPoint (Компанія TriPath Imaging). Її завданням є визначення наявності ракових клітин за взятою у пацієнта пробою. Спочатку творці системи сформувавши за допомогою експертів набір основних ознак, за якими відрізняються клітини, що відхилилися від норми. Далі система проходить навчання, аналізуючи проби, для яких вже був поставлений діагноз. Однак після того як систему випускають, функції навчання в ній блокуються, оскільки це може привести до того, що недосвідчений лікар може погіршити якість системи, що є неприйнятним. Система щорічно виробляє аналіз 5 млн зображень клітин, що становить 10% від всіх подібних аналізів в США.

На сьогодні має місце переважаюче виробничих ЕС і бізнес-додатків. При цьому виразний ріст спостерігається в області фінансових ЕС. Складно переоцінити значення фінансових експертних систем для бізнесу.

Фінансові ЕС системи активно використовуються в комерційних підприємствах - для аналізу власного економічного, фінансового, бухгалтерського, інвестиційного, інноваційного станів, для аналізу рентабельності, складання звітів перед власниками підприємства і утримувачами

акцій або для тактичного і стратегічного планування виництва і збуту.

Головна відмінність ІС і ЕС від інших програмних засобів - це наявність бази знань (БЗ), де знання зберігаються в формі, зрозумілою фахівцям предметної області, і можуть бути змінені і доповнені також в зрозумілій формі.

Це і є мови представлення знань - МПЗ.

До останнього часу саме різні МПЗ були центральною проблемою при розробці ЕС. Зараз існують десятки мов або моделей представлення знань.

Найбільшого поширення набули наступні моделі:

- 1) продукційні правила;
- 2) семантичні мережі;
- 3) фрейми FRL;
- 4) логіка першого порядку;
- 5) об'єктно-орієнтовані мови (наприклад, CLOS — Common Lisp Object System).

Для перерахованих вище моделей існує відповідна математична нотація, розроблені системи програмування, що реалізують ці МПЗ, і є велика кількість реальних комерційних ЕС.

У нас розробок в області ЕС ведеться небагато, але є певний інтерес серед найбільш «просунутих» і кваліфікованих шарів фахівців - фінансистів, топ-менеджерів, викладачів, інженерів, медиків, психологів, програмістів, лінгвістів.

В останні роки цей інтерес має поки досить слабе матеріальне підкріплення - реальна нестача грамотних фахівців, обмежене фінансування досліджень в цій області, слабкий вітчизняний ринок програмних продуктів для розробки ЕС.

Тому триває спекуляція термінології, з'являється можливість поширення «підробок» під експертні системи у вигляді численних діалогових систем та інтерактивних пакетів прикладних програм, які дискредитують в очах користувачів цей перспективний напрямок.

Найбільші труднощі в розробці ЕС викликає сьогодні не процес машинної реалізації систем, а домашнього етапу аналізу знань і проектування бази знань. Цим займається спеціальна наука - інженерія знань.

Сучасному стану цієї науки і присвячені наступні лекції.