

**РОЗРОБКА ТЕХНОЛОГІЇ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ  
ЗНАНЬ НА ОСНОВІ МЕТОДІВ ЕКСПЕРТНИХ СИСТЕМ**

*В даній статті розглянуто проблеми застосування експертних систем в області контролю знань. Виділено базу знань експертної системи як основний концептуальний елемент контролю знань. Структурно така база знань описується на основі об'єктів, правил, доменів, області задач, екранних форм та функцій.*

*In the given article the problems of applying of consulting models are reviewed in the field of the control of knowledge. The knowledge base of the consulting model as the basic conceptual member of the control of knowledge is allocated. Structurally such knowledge base is described on the basis of objects, rules, domains, area of problems, screen shapes and functions.*

Ключові слова: експертна система, контроль знань.

Одним з пріоритетних напрямів розвитку України є система вищої освіти в якій активно впроваджуються елементи та методи Болонського процесу. Тому важливим на даному етапі є питання комп'ютеризації контролю знань. Існуючі системи тестового контролю призначені в основному для оцінки кількісних показників рівня знань. Вони в випадковому або наперед визначеному порядку генерують ряд тестових питань, при цьому, кожна конкретна відповідь додає до загального результату певну суму балів. Проте така оцінка як правило не є якісним показником рівня знань, так як вона носить статистичний характер і по своїй суті є рейтинговою. Такий підхід до контролю знань жодним чином не вирішує завдання повноти та комплексності процесу, оскільки послідовність генерації тестових питань не залежить від відповідей студента на попередні питання. Від цих недоліків позбавлені системи адаптивного тестування, основною структурною формою реалізації яких виступають експертні системи.

Експертні системи мають різноманітні сфери застосування, зокрема це діагностика, прогнозування, планування, контроль, управління та навчання. Для вирішення проблем контролю знань використовується клас навчальних експертних систем. Їхнє призначення полягає в виведенні сумарної характеристики знань, формуванні умінь навчальної і практичної діяльності та забезпечення необхідного рівня засвоєння, що встановлюється зворотнім зв'язком.

Питання автоматизованого контролю знань розглядається в багатьох дослідженнях. Так в [1] пропонується включити в автоматизовану систему контролю знань такі основні можливості як: автономна розробка тестових завдань; налагодження в діалоговому режимі параметрів системи; можливість переходу від виду введення відповідей з клавіатури до варіанту вибору відповідей зі списку; автоматизоване заповнення бази даних тестів з текстових файлів. Про те в даному дослідженні описуються лише основні принципи автоматизованих систем тестування, але не розглядається питання адаптивного тестування. При аналізі алгоритмів оцінки знань в роботі [2] вказується на те, що не існує універсального алгоритму, і тому потрібно передбачити можливість відключення або зміни певних параметрів системи тестування, тобто пропонується використання адаптивного алгоритму тестування. Використання адаптивних автоматизованих систем тестування розглядається також в роботах [3,4]. Про те тільки в [5] пропонується використання експертних систем в сфері автоматизованого контролю знань з метою найбільш точного визначення рівня знань.

Водночас мало дослідженим залишається питання побудови ефективних баз знань експертних систем контролю знань. Отже метою даного дослідження є введення методів та засобів експертних систем для контролю якості знань.

Представлення знань в проектованій нами експертній системі контролю знань базується на таких елементах як: об'єкти (*Object*), правила (*Rule*), домени (*Scope*), задачі (*Task*), форми (*Form*), функції (*Function*).

Всі входження бази знань створюються за допомогою налаштувань в опціях системи, які активуються утилітою *Editor*. Вікно даної утиліти зображено на рис. 1.

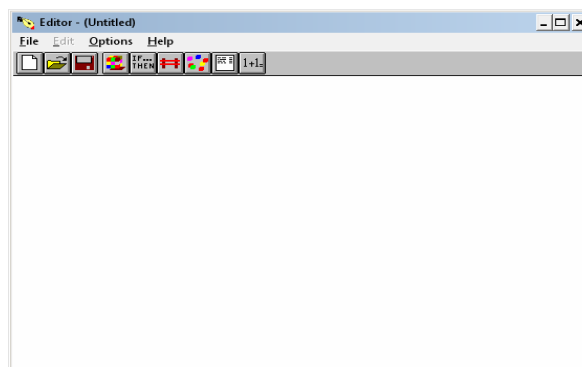


Рис. 1. Вікно утиліти Editor.

Компонент об'єкти (*Object*) служить для збереження інформації про об'єкти контролю знань які використовуються в системі (рис. 2).

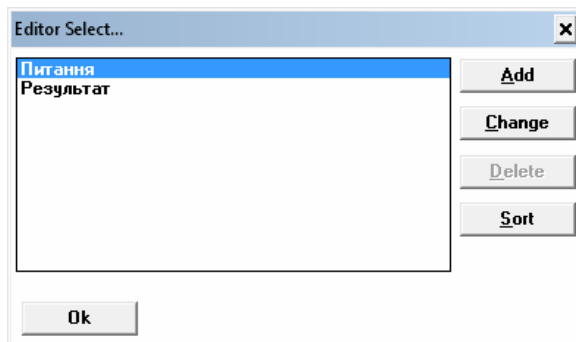


Рис. 2. Діалог вибору об'єктів контролю знань.

Він складається з імені (*name*) та атрибутів (*attributes*), які в свою чергу можуть бути одного з трьох типів: логічний (*boolean*), числовий (*numerical*) та перераховуваний (*enumerated*). Атрибути додаються в об'єкт за допомогою кнопки додати (*Add*) (рис. 3).

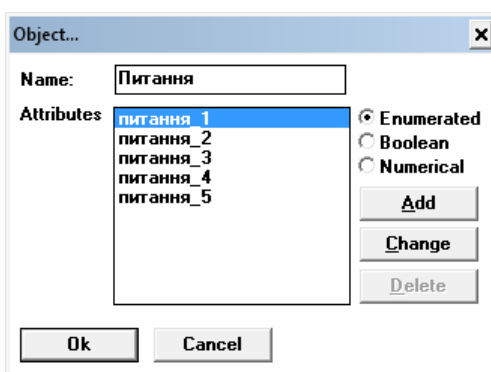


Рис. 3. Діалог вибору об'єкта для редагування.

Логічний (*boolean*) атрибут може приймати одне з 3-ох значень: *TRUE*, *FALSE*, *UNKNOWN*, відповідне відображення яких в режимі консультації задається в блоці відповіді (*Answers*).

Атрибут має назву (*name*), розшифровку (*transcript*) та стрічку підказки під час консультації (*prompt*). Підказку (*prompt*) користувач бачить в режимі консультації, а розшифровка даної підказки використовується для формування питання на природній для користувача мові (рис. 4).

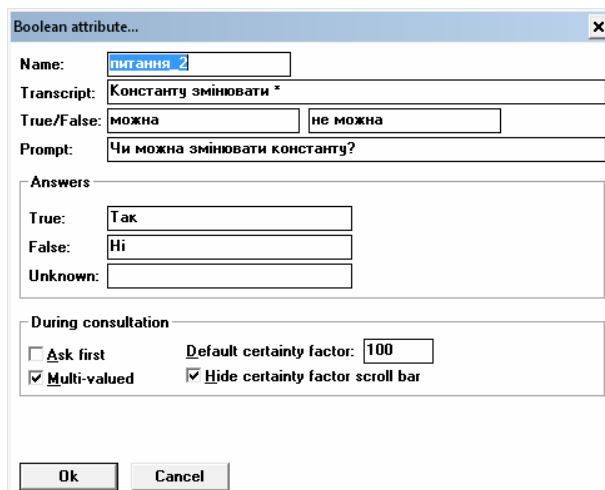


Рис. 4. Представлення логічного атрибуту.

Числовий (*numerical*) атрибут зберігає числові значення заданого об'єкта. Для нього можна задати такі значення: найменше (*Lower*), найбільше (*Upper*) і по замовчужанню (*Default*). Як і для логічного типу, він ідентифікується назвою (*Name*), розшифровкою (*Transcript*) та підказкою (*Prompt*), які використовуються аналогічно попереднім атрибутам (рис. 5).

Рис. 5. Представлення числового атрибуту.

Атрибут перерахування (*enumerated*) використовується для занесення в базу знань інформації про об'єкт у вигляді послідовності елементів.

Назва (*Name*), розшифровка (*Transcript*), та підказка (*Prompt*) використовуються для тих самих цілей, що і в попередніх атрибутах. В поле опції (*Options*) з допомогою кнопки додати (*Add*) заносяться елементи перераховуваного типу (рис. 6).

Рис. 6. Представлення перераховуваного атрибуту.

В діалозі налаштування атрибутів є окремий блок опцій контролю режиму виконання (*During consultation*). Якщо опція отримання значення для атрибута (*Ask first*) не активована і не заповнена стрічка підказки під час консультації (*prompt*), то експертна система робить вивід значення для властивості не вимагаючи його від користувача. Якщо дана опція не активована, але введена підказка для властивості, то система спершу пробує вивести значення для властивості. Якщо це значення є не правильним, то використовується підказка (*Prompt*) яка є значенням користувача. У випадку коли опція значення для атрибута (*Ask first*) є активована, експертна система запитує користувача значення для властивості. Якщо користувач не знає відповіді, тоді виникає спроба вивести це значення. В цьому випадку повинна бути заповнена стрічка підказки (*Prompt*). Коли активована опція *Multi-valued* то, властивість може мати більш ніж одне значення протягом консультації. Якщо знайдено більше одного значення для однієї і тієї ж властивості, то ці значення належать екземпляру об'єкта з тією ж властивістю. В іншому випадку властивість може приймати тільки одне значення. Для кожної властивості виконується пошук значення, і створюється новий екземпляр об'єкта з такою ж властивістю.

Вкладка правила (*Rule*) утиліти *Editor* використовується для зв'язування об'єктів (*Object*) між собою (рис. 7).

Рис. 7 Діалог вибору правил.

В даному діалоговому вікні можна створити нове правило та видалити або відредагувати уже існуюче. Прикладом типового правила в проєктованій системі контролю знань є:

```

RULE_1 (MAIN SCOPE)
IF Ідентифікатор це: ім'я змінної
AND Константу змінювати не можна
AND В 1 Byte кількість Bit = 8
AND Оператор присвоєння в C++ позначається: =
AND в кінці рядка коду a=2 в C++ ";" потрібна
THEN ASSERT результат тестування наступний: тестування пройдено (100 %)
  
```

За допомогою кнопки сортувати (*Sort*) можна відсортувати правила в алфавітному порядку.

Кожне правило характеризується коефіцієнтом впевненості (*CF – Certainty Factors*) (рис. 8) який вказує спосіб застосування правила при істинній передумові.

Кожне правило міститься в певному домені (*Scope*). Правило може використовуватися тільки тоді коли його домен активований. Вхідження в передумовах правил можуть з'єднуватися з допомогою логічних операцій «і» (*And*) чи «або» (*Or*). Коли вхідження з'єднуються за допомогою *And* вони повинні бути істинними одночасно для правила яке є активним, коли вони з'єднуються через *Or* достатньою є істинність тільки хоча б одного з вхіджень.

Правила специфікуються по типу виконання (*Actions*) (режими *assert, replace, retract, activate*).

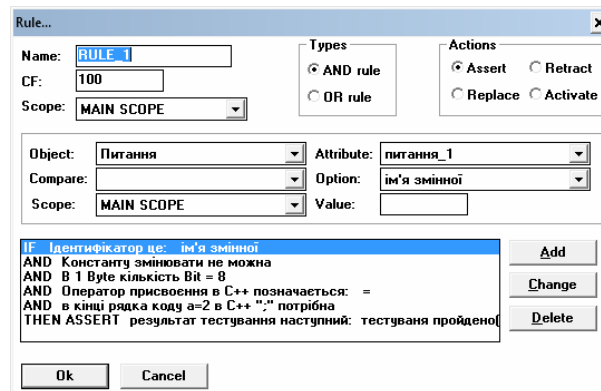


Рис. 8. Діалог редагування правил

Коли активовано опцію заявка (*Assert*) тоді завершене правило поміщається в базу знань. Під час активації опції заміна (*Replace*) завершене правило також поміщається в базу знань, але коли в базі вже є факт про властивість, цей факт видаляється з бази знань. Якщо це перший факт то властивість, заміна (*Replace*) поводиться ідентично властивості заявка (*Assert*). Опція витягування (*Retract*) видаляє факт, який відповідає за завершення правила в базі знань. Якщо такого факту не знайдено, то нічого не відбувається. Правило використовується в процесі висновку, тільки коли його домен активний (*Active*). Головний домен бази знань завжди активний, інші домени активізуються правилами.

Умова формується за допомогою об'єкта (*Object*), компаратора (*Compare*), домена (*Scope*), атрибуту (*Attribute*), вибору (*Option*) або значення (*Value*) у випадку числової властивості. Умова може складатися з довільної кількості передумов. Числові атрибути заносяться в поле *Value*, а логічні та перераховувані в поле *Option*.

Правила активуються в доменах (*Scope*) (рис. 9).

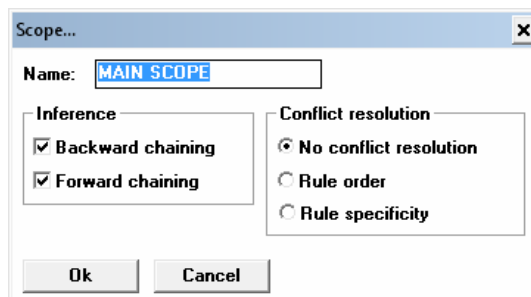


Рис. 9. Діалог налаштування домену.

Головний домен по замовчуванню створюється в *Editor*, але його за необхідності можна змінити. Правило може бути виконане тільки за умови що його домен активний. Головний домен завжди активний, тоді як інші домени активуються правилами. За допомогою доменів можна поділити базу знань на окремі частини в яких відбувається управління типом висновку і вирішення протиріч. Домен ідентифікується

назвою. В діалозі налаштування доменів є блок опцій вивід (*Inference*) і вирішення протиріч (*Conflict resolution*).

За допомогою діалогу задачі (*Task*) відбувається вибір вжоджень. Даний діалог дозволяє задати ім'я (*Name*) задачі та вибрати об'єкт (*Object*) і атрибут (*Attribute*) які будуть використовуватись в формуванні висновку. Задача може містити одне або більше вжоджень, які будуть первинною метою в процесі висновку (рис. 10).

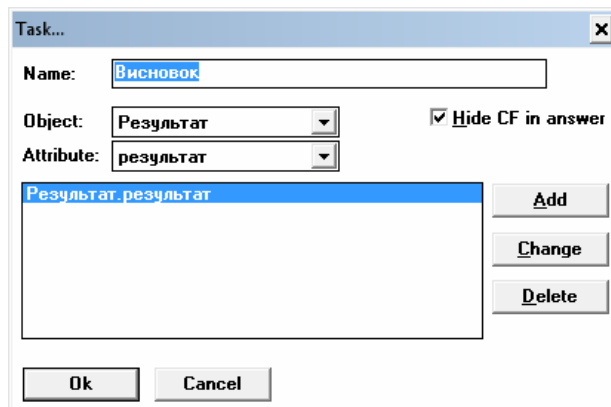


Рис. 10. Діалог представлення задач.

Діалог форма (*Form*) містить список питань з типовими відповідями (рис. 11). Він дозволяє сформулювати питання та відповіді на них які будуть запропоновані користувачеві під час проходження консультації.

Для створення форми необхідно задати її назву (*Title*) та заповнити поля *Object*, *Attribute* та *Default*. Поле *Default* дозволяє вказати значення яке буде використовуватись під час консультації по замовчуванню.

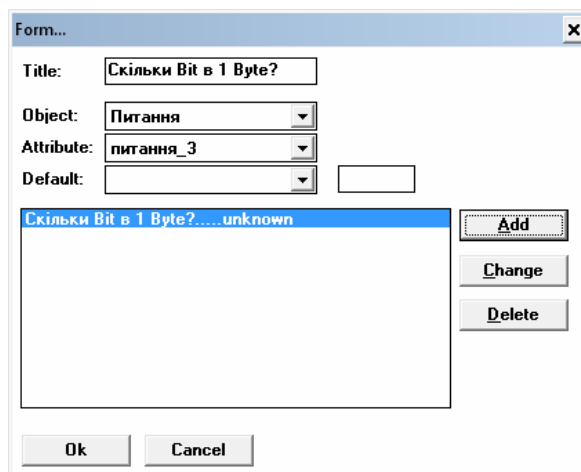


Рис. 11. Діалог представлення форм.

Функції (*Function*) описуються за допомогою формату RPN (тобто, змінні і константи описуються перед операторами).

Обчислення може містити константи і числові значення. Додавання атрибутів в функцію відбувається за допомогою кнопки додати (*Add*). Операції додавання (*Plus*), віднімання (*Minus*), множення (*Multiply*), ділення (*Divide*) додаються вибором відповідних ім полів та натисканням кнопки *Add* (рис. 12).

Інтерфейс проектованої експертної системи дозволяє виконання таких базових опцій:

- *Open* – відкривати вже існуючу базу;
- *Reload* – перезавантажити базу знань;
- *Setup* – налаштування користувача та активація автоматичного завантаження бази знань;
- *Exit* – вихід з системи.
- *Consult* – за допомогою даного пункту меню можна здійснювати процес консультування а також працювати з базою знань.
- *Start* – початок процесу консультації;

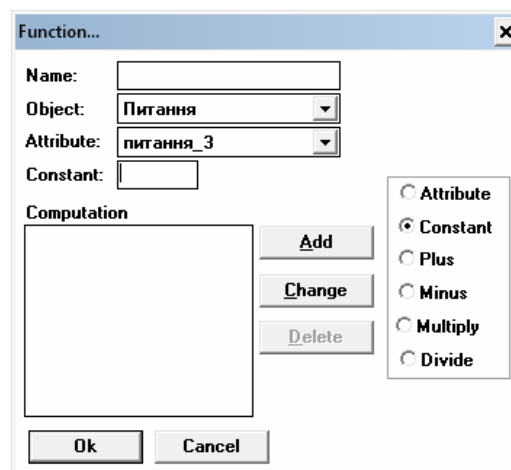


Рис. 12. Діалог використання функцій

- *Continue* – продовження процесу консультування;
  - *Tools* – утиліти для розробки бази знань, її редагування та відлагоджування.
  - *Agenda* – інструмент для перегляду активних параметрів та можливих варіантів відповіді на них;
  - *Blackboard* «чорна дошка» – використовується для запам'ятовування процесу консультації. Сюди записуються факти, які були отримані в процесі консультації з користувачем;
  - *Trace* – відлагоджування програми, може відслідковувати цілі та порядок набування тверджень з певних гіпотез;
  - *Editor* – вбудований в систему редактор для створення та редагування баз знань;
  - *Help* – допомога по експертній системі контролю знань.
  - *Contents* – опис системи;
  - *Search for help on* – пошук допомоги за ключовими словами;
  - *How to use help* – допомога по користуванні допомогою;
- Після завантаження бази знань в систему з'явиться інформація про неї (рис. 13):

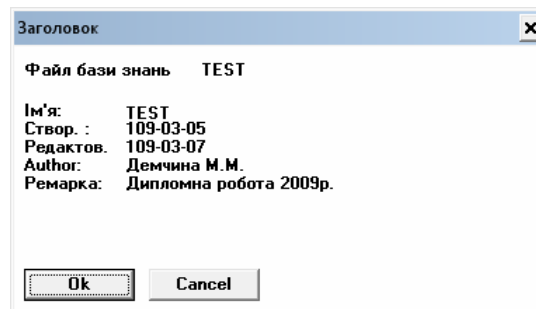


Рис. 13. Властивості бази знань

Щоб розпочати консультацію потрібно вибрати пункт меню *Consult->Begin*. Після цього система буде задавати питання, на які повинно давати відповіді (рис. 14). Питання в даній системі можуть бути 4 типів: перераховувані, логічні, числові та текстові. Тому вигляд запитань, які система буде задавати будуть відрізнятися залежно від їх виду.

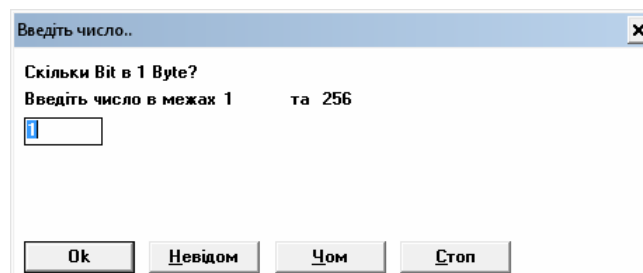


Рис. 14. Діалогове вікно експертної системи

Після того як всі питання з пакету було задано експертна система генерує висновок про успішність складання даного тесту (рис. 15).

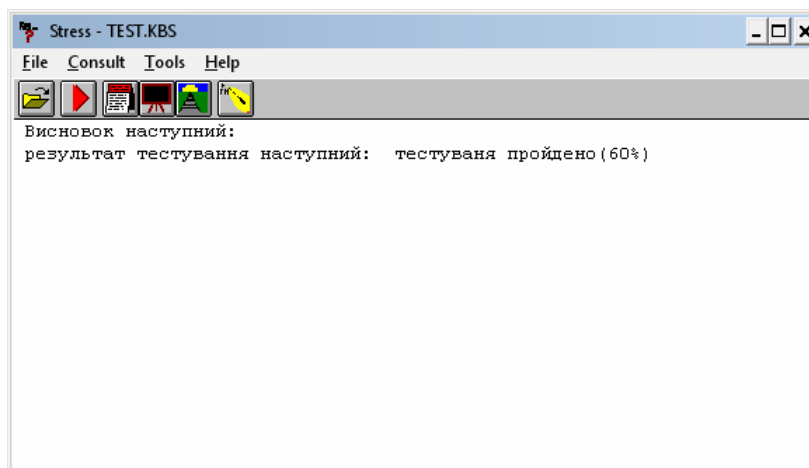


Рис. 15. Висновок експертної системи

**Висновки**

Таким чином, використання запропонованої автоматизованої системи контролю знань на основі методів експертних систем дозволяє з більшою достовірністю визначати якісний рівень знань студентів і генерувати керуючі впливи для подальшого вдосконалення структури та представлення знань в предметній області.

Подальші дослідження даного напрямку будуть зосереджені на формалізації структури представлення знань інформаційної інтелектуальної системи для предметної області «інформаційні та комп'ютерні технології».

**Література**

1. Богданевич М.А. Організація системи перевірки знань для поточного та підсумкового контролю. Друга Міжнародна науково-методична конференція «ІНТЕРНЕТ – ОСВІТА – НАУКА – 2000», м. Вінниця.
2. Зайцева Л.В., Прокофьева Н.О., Куплис У.Г. Компьютерные системы в дистанционном обучении // ТЕЛЕМАТИКА'2001 – Санкт-Петербург, 2001. – С. 109-111.
3. Паволоцкий А.В. Методика проведения автоматизированного контроля знаний // Доповідь – МПДУ, 2007
4. Пустынникова И.Н. Методология конструирования диагностирующей экспертной системы (на базе оболочки BESS) // Вісник Донецького університету. – Серія А. Природничі науки. – 1998. – № 1. – С. 182-187.
5. Андреев А.Б., Усачев Ю.Е. Экспертная система анализа знаний как инструмент контроля усвоения зачетных единиц // Проблемы введения системы зачетных единиц в высшем профессиональном образовании: Материалы к Всероссийскому совещанию 23 апреля 2003 года, г. Москва / Под ред. В.Н. Чистохвалова. – М.: Изд-во РУДН, 2003. – 100 с.

Надійшла 5.11.2009 р.

УДК 621.37:681.33

І.І. МІТАСОВ, О.В.РУБАН, В.М. РУДНИЦЬКИЙ  
Хмельницький національний університет

**ОСОБЛИВОСТІ ЗАСТОСУВАННЯ МЕТОДІВ СТИСНЕННЯ ДАНИХ**

*В статті охарактеризовані відомі методи стиснення даних, розглянуті особливості стиснення статистичними і словниковими методами, з втратами і без втрат, порівняно статичні і динамічні моделі. Описані можливості архіваторів.*

*In article were discussed methods of different data compression algorithms. Different archivers and their capabilities were discussed.*

Ключові слова: стиснення даних, алгоритми стиснення даних.

**Вступ**

При передачі та зберіганні інформації завжди виникає проблема розміру повідомлень. Розвиток засобів комп'ютерної техніки веде до збільшення швидкостей передачі та об'ємів накопичувачів. Може скластись враження про недоцільність використання методів стиснення інформації через це зростання. Але це погляд лише з одного боку. З іншого боку відбувається зростання обсягів самих інформаційних повідомлень. При чому не рідко зростання обсягів передачі відбувається швидше, ніж зростають швидкості передачі. Окрім цього за передачу інформації потрібно платити. Зменшення плати за передачу інформації ще одна з позитивних сторін використання методів стиснення інформації. Тому проблема стиснення інформації завжди залишиться актуальною.

Мета стиснення – зменшення кількості біт, необхідних для зберігання або передачі заданої інформації. Це дає можливість передавати повідомлення швидше і зберігати їх більш економно і оперативно. Останнє означає, що операція отримання даної інформації з пристрою її зберігання проходить швидше, а це можливо, якщо швидкість розпакування даних вище за швидкість прочитування даних з носія інформації. Стиснення дозволяє, наприклад, записати більше інформації на дискету, “збільшити” розмір жорсткого диска, прискорити роботу з модемом і т.п. При роботі з комп'ютерами широко використовуються програми-архіватори даних формату ZIP, GZ, ARJ та інших. Методи стиснення інформації були розроблені як математична теорія, яка довгий час (до першої половини 80-х років), мало використовувалася в комп'ютерах на практиці.

Збільшення обсягів передачі, статичних картинок, звуку та відеозображень призвело до розроблення багатьох методів стиснення з втратами або незворотного кодування. Для передачі саме цих типів повідомлень допускається використання таких методів. Але досить часто необхідно передавати документи, фрагменти баз даних, виконувати файли. Неможливість повного відновлення таких файлів призведе фактично до втрати інформації.