

УДК 683.1

О.Б. Вовк, А.Я. ВовчинаНаціональний університет “Львівська політехніка”,
кафедра “Інформаційні системи та мережі”

ПІДХІД ДО ОЦІНКИ ЯКОСТІ ВРАХУВАННЯ ЧАСОВОГО ФАКТОРА В ІНФОРМАЦІЙНИХ СИСТЕМАХ НА ОСНОВІ РЕЛЯЦІЙНИХ БАЗ ДАНИХ

© Вовк О.Б., Вовчина А.Я., 2003

Formal approaches to defining informational systems quality are considered in this article. Here a rating scale of correlation criteria with the general utility of the Informational Systems is offered. The evaluation of the metrics criteria facilities with the possibility to assign appropriate defining algorithms is determined. Consideration of the time parameter, general characteristic which is suggested for informational systems quality evaluation. The sample of this factor figure system realization by the example of the informational system in a museum establishment here is presented.

Розглянуто формальні підходи до визначення характеристик якості інформаційних систем. Запропоновано шкалу оцінки кореляції критеріїв з загальною корисністю ІС. Визначено оцінку можливості задання метрик критеріїв з можливістю задання відповідних алгоритмів визначення. Запропоновано для оцінки якості інформаційної системи таку загальну характеристику, як врахування часового параметра. Наведено приклад реалізації системи показників цього фактора на прикладі інформаційної системи музейної установи.

ВСТУП. ЗАГАЛЬНА ПОСТАНОВКА ПРОБЛЕМИ

Попередні дослідження, результати яких є предметом розгляду, полягали в тому, щоб виявити необхідну сукупність характеристик якості програмного забезпечення (ПЗ) й встановити потім для кожної такої характеристики один або більше вимірних показників. Це дозволило б стверджувати, якою мірою той чи інший програмний продукт має деяку властивість, і, по-друге, давало б можливість виводити загальну оцінку його якості як функцію значень власних показників. Більшість досліджень у цій сфері рідко розглядають інформаційні системи (ІС), основою яких, як правило, є база даних [11]. Для більш повної загальної оцінки якості інформаційної системи необхідно враховувати якість бази даних. Можна було б формалізувати підхід до визначення критеріїв якості інформаційних систем на базі реляційних баз даних, але такий широкий напрямок не є предметом дослідження цієї роботи. Доцільно розглянути таку важливу загальну характеристику інформаційних систем, як якість опрацювання часового фактора.

Розглянемо спочатку загальний підхід до визначення якості програмного забезпечення.

АНАЛІЗ СУЧАСНИХ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Формування показників якості ПЗ виходить з таких вихідних положень [7, 8, 12].

Перше з них полягає в тому, що при розробці програмного забезпечення важливішим є *де* і *як* виявляються його дефекти, ніж *частота* їхньої появи. Тому найбільшу цінність представляють такі автоматичні засоби аналізу якості програмного забезпечення, що реєструють конкретні недоліки, а не просто вказують на них.

Друге вихідне положення полягає в тому, що практично для будь-якої простої формули кількісної залежності можна підшукати приклади, що спростовують або ставлять під сумнів прийнятність її як критерію якості програмного забезпечення. Наприклад, існує міра структурованості програми, що обчислюється як середня довжина програмних модулів. Однак припустимо, що ми маємо програмне забезпечення в складі n керуючих програм обсягом по 100 операторів кожна і бібліотеку з m обчислювальних програм, що містять усього по 5 операторів. Така система програмного забезпечення повинна визнаватися добре структурованою при будь-яких прийнятних значеннях m і n . Проте, якщо $n=2$ і $m=98$, середня довжина модуля дорівнює 6,9 оператора, а при $n=10$ і $m=10$ вона становить 52,2 оператора, що призводить до неправильних висновків про ступінь структурованості.

Третє положення полягає в тому, що сама технологія розробки програмного забезпечення поки ще знаходиться в стані швидкого еволюційного розвитку, що в деяких випадках ускладнює використання яких-небудь стійких показників якості. Введення строгих кількісних метрик означало б примусовий вплив на існуючі методи роботи і могло б дати небажані результати.

Четверте вихідне положення зводиться до того, що обчислення і розуміння конкретного числового значення деякого єдиного узагальненого показника якості програмного забезпечення може виявитися зв'язаним з такими труднощами, які не можна виправдати. Основною проблемою тут є те, що багато індивідуальних характеристик якості програмних засобів є суперечливими. Наприклад, збільшення ефективності нерідко стає можливим лише за рахунок погіршення зрозумілості структури ПЗ. Коли ж виникають подібні конфліктні ситуації, користувачі зазвичай не можуть вказати, які ж характеристики з їхнього погляду є більш істотними [6].

Отже, необхідним є створення сукупності взаємопов'язаних характеристик якості програмного забезпечення з подальшою розробкою системи показників, що дозволятимуть виявляти аномалії відповідних характеристик.

Необхідними на цьому етапах є:

1. Встановлення сукупності характеристик, важливих з погляду забезпечення високої якості ПЗ, що не перекривають одна одну і є достатньо вичерпні.
2. Розробка можливої системи показників для оцінки ступеня, у якому конкретному програмному забезпеченню властива визначена характеристика.
3. Дослідження встановлених характеристик і зв'язаної з ними системи показників з метою з'ясування ступеня їхньої кореляції з якістю програмного забезпечення, важливості застосування або переваг використання, можливості представлення в кількісній формі і легкості автоматичного отримання.
4. Оцінка кожного з обраних показників з позицій зазначених вище критеріїв, а також з погляду його взаємодії з іншими показниками з метою з'ясування перекриттів, залежностей, недоліків тощо.
5. Перетворення безлічі вихідних характеристик, установлених на етапі 1, у сукупність більш незалежних і вичерпних характеристик якості програмного забезпечення шляхом уточнення вихідної множини на основі оцінок, отриманих на етапі 4.
6. Уточнення системи показників якості і їхнє перегрупування відповідно до переглянутого набору характеристик.

В [8, 10] введені такі характеристики якості програмного забезпечення як: зрозумілість, завершеність, осмисленість, узгодженість, зручність експлуатації, оцінюваність, надійність, структурованість, ефективність.

На другому етапі визначають вимірні властивості (тобто їх метрики), що могли б служити корисними індикаторами її зрозумілості, зручності експлуатації тощо. Тоді будь-яка міра зрозумілості програми виявиться одночасно і мірою зручності її експлуатації, оскільки для підтримки програмного забезпечення в працездатному стані необхідно, щоб воно було зрозумілим обслуговуючому персоналові. Водночас деякі з показників зручності експлуатації не впливають на зрозумілість програми. Наприклад, такі засоби забезпечення оцінюваності, як проміжний вивід і контроль даних, що вводяться, важливі для проведення тестування в процесі експлуатації, але ніяк не зв'язані з поняттям зрозумілості.

Так утворюється деревоподібна структура характеристик програмного забезпечення, фрагмент якої показаний на рис. 1.

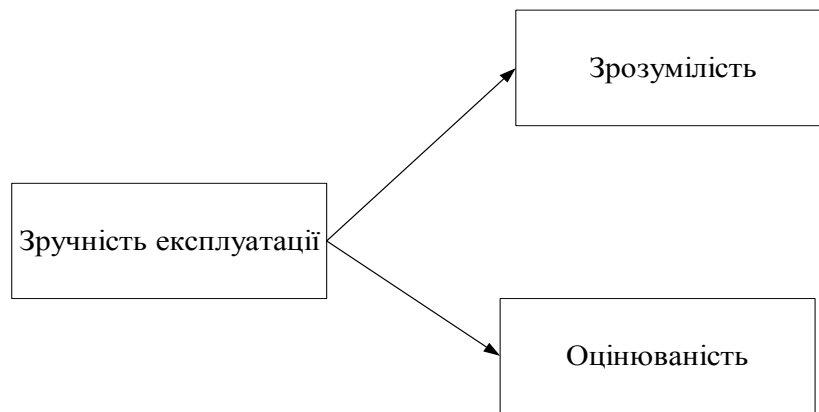


Рис. 1. Приклад взаємозв'язку характеристик програмного забезпечення

На рис. 1 стрілки вказують логічне відношення проходження: якщо програма зручна в експлуатації, то вона обов'язково зрозуміла й оцінювана. Інакше кажучи, високий ступінь зручності експлуатації (експлуатаційної надійності) вимагає високого ступеня зрозумілості й оцінюваності програмного забезпечення.

Існує ще один рівень більш простих показників, які, якщо розглядати рис. 1, повинен знаходитися праворуч від характеристик зрозумілості й оцінюваності. Наприклад, якщо програмне забезпечення зрозуміле, воно обов'язково буде структурованим, погоджене й осмислене (три характеристики, що входять у вихідну множину), а, крім того, відкритим і інформативним (дві додаткові характеристики, що не перекриваються з першими трьома). Розширивши в такий спосіб множину вихідних характеристик, сукупність характеристик може бути відображена у вигляді дерева, у якому більш елементарні характеристики є необхідною умовою існування більш узагальнених.

В остаточному вигляді дерево характеристик якості ПЗ можна представити рис. 2.

Верхні рівні даної структури показують сфери застосування оцінок якості програмного забезпечення. Так, наприклад, вихідна корисність, зручність експлуатації є необхідними (але не достатніми) елементами оцінки загальної корисності. Властивість вихідної корисності вимагає, щоб програмне забезпечення було надійним, достатньо

ефективним і розробленим з урахуванням людино-машинного спілкування, однак воно зовсім не вимагає, щоб користувач випробував його, розуміючи внутрішні механізми роботи, міг модифікувати або спробувати застосовувати програмне забезпечення не в сфері його основного призначення. Для зручності експлуатації програмного засобу необхідно, щоб користувач розумів його, міг модифікувати і випробувати, спираючись на можливості, надані з урахуванням людського фактора, але зовсім не потрібно наявності високої надійності, ефективності.

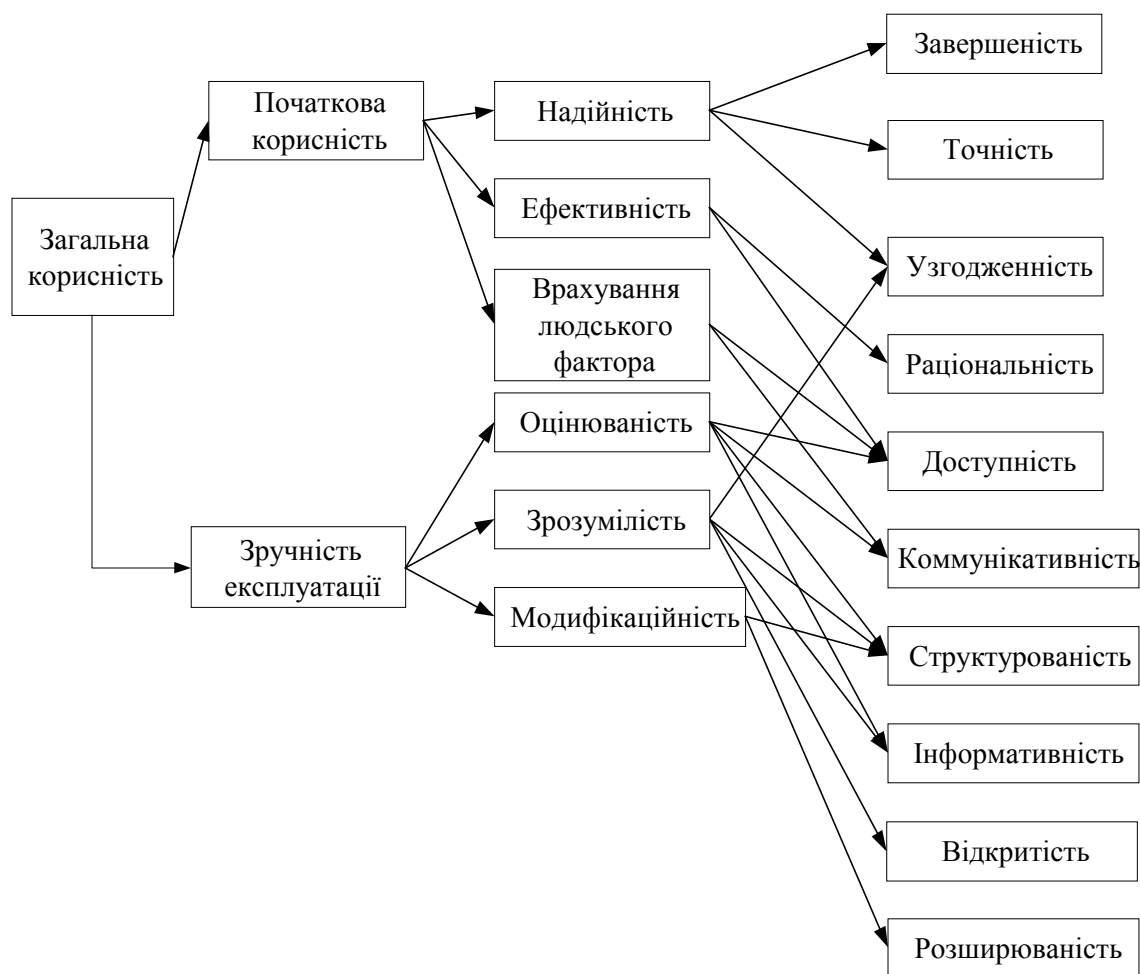


Рис. 2. Дерево характеристик якості програмного забезпечення

Характеристики, розташовані на нижньому рівні дерева, являють собою множину елементарних властивостей, що суттєво відрізняються одна від однієї і групуються в набори відповідно до умов, необхідних для існування конкретних характеристик проміжного рівня.

Виділення множини елементарних характеристик, наведених на рис. 2, створює більш ґрунтовну основу для визначення кількісних показників, що могли б застосовуватися для оцінки ступеня, в якому програмне забезпечення володіє як зазначеними елементарними властивостями, так і характеристиками більш високих рівнів. І ті, й інші повинні брати участь у формуванні оцінки загальної корисності програмного забезпечення (оцінки вищого рівня).

ВИМІРЮВАННЯ ЯКОСТІ ІС

Термін “показник” визначається як міра ступеня, у якому деякий продукт (у даному випадку інформаційна система) володіє тією або іншою характеристикою і вона властива їй (у даному випадку якість) у процесі експлуатації. Зазначимо, що процеси розробки й уточнення показників і характеристик нерозривно зв’язані і мають ітеративний характер. Сформульовані і проаналізовані показники, застосовані для оцінки програмного забезпечення, в результаті кількох ітерацій привели до побудови наведеної вище ієрархічної сукупності характеристик якості програмного забезпечення [7].

На основі цієї структури можна оцінити корисність кожного з її елементів за такими критеріями.

1. *Наявність кореляції якості програмного забезпечення.* Перевірка за цим критерієм передбачає одержання відповіді на запитання, чи дійсно показник, що за припущенням може бути мірою “елементарної” характеристики, пов’язаний з якістю програмного забезпечення? При цьому під високою позитивною кореляцією розуміється (грубо) ситуація, коли більшість програмного забезпечення з високими оцінками за цим показником мають одночасно і відповідну “елементарну” характеристику. Зрозуміло, можна використати і більш суворе в статистичному сенсі визначення кореляції, однак розглянутий етап формування оцінок має досить суб’єктивний характер, при якому навряд чи є виправдане більш точне обчислення коефіцієнтів кореляції, що вимагає значного часу для збирання великого обсягу даних за багатьма різноманітними інформаційними системами. З врахуванням цих міркувань для показника можна встановити таку шкалу оцінок:

З – дуже висока позитивна кореляція з якістю; майже всі інформаційні системи з високим значенням даного показника повинні мати зв’язані з ним характеристики (*завжди*).

МЗ – висока позитивна кореляція з якістю; значна частина (75—90 %) усіх ІС з високим значенням даного показника повинна мати відповідну характеристику (*майже завжди*).

П – ІС з високим значенням даного показника *переважно* (у 50—75% випадків) будуть мати відповідну характеристику якості.

І – деякі з програмних засобів з високим значенням даного показника будуть мати зв’язані з ним характеристики (*іноді*).

2. *Потенційна вигода застосування показника.* Одні показники дають дуже важливу для розуміння властивостей програмного забезпечення інформацію і допомагають приймати рішення як його розробникам, так і майбутнім користувачам; інші ж, хоча і подають цікаву інформацію з погляду виявлення згодом можливих утруднень, не приводять до великих втрат при невисоких значеннях навіть у випадку сильної кореляції цих показників з відповідними характеристиками якості програмного забезпечення. Зрозуміло, що твердження про потенційну вигоду показника залежить від цілей використання ІС, тому для її оцінки було встановлено таку шкалу:

5 – вкрай важливо, щоб цей показник мав високе значення, інакше можливі значні утруднення;

4 – важливо, щоб цей показник мав високе значення;

3 – добре б мати високе значення цього показника;

2 – деякою мірою корисно мати високе значення цього показника;

1 – виграш від високого значення цього показника вкрай незначний; при низьких значеннях відсутніх втрат немає.

3. *Можливість кількісного представлення показника і зручність його автоматичної оцінки.* Можливе досягнення деяким показником найвищих значень як за кореляцією з якістю програмного забезпечення, так і відносно корисності застосування, однак при цьому визначення числової його величини може вимагати великих витрат часу і коштів. Якщо оцінка показника вимагає оцінювання програмного забезпечення експертом і винесення останнім експертного ствердження, то числове значення показника буде зазвичай нести в собі набагато менше інформації, ніж одержить фахівець-експерт у процесі оцінювання. Крім того, показники, що вимагають експертизи, вкрай дорогі для використання на практиці. Отже, для великих ІС бажано мати алгоритм автоматичної оцінки, що аналізував би програмне забезпечення і формував би відповідне значення показника (а якщо можливо, і видавав би на друк перелік тих місць програмного забезпечення, у яких відповідна характеристика якості відсутня). Часто виявляється більш доступним проміжний варіант, при якому використовується автоматичний компаратор, що порівнює властивості реального програмного забезпечення з переліком бажаних характеристик, складеним користувачем. Для оцінки доцільного способу квантифікації показника, тобто способу, що забезпечує найбільш економічне його визначення, можна застосувати таку шкалу:

АА – потрібен *автоматичний алгоритм*;

КО – можливо використання *компаратора* при наявності заданого переліку необхідних характеристик;

НС – потрібен *некваліфікований фахівець* для оцінювання програмного забезпечення або особа без великого досвіду;

КС – потрібний *кваліфікований фахівець*;

ЕК – потрібен *експерт*;

ПР – потрібен *“прогін”* програмного забезпечення.

Автоматизація таких процесів формування оцінок або контроль наявності визначених інформативних зведень можуть бути виконані досить легко і просто. Водночас автоматизація таких оцінкових операцій, як глобальне сканування ІС для виявлення помилок і гарантування незмінності їхніх компонент між повторними звертаннями до них хоча і можлива, але складна. Є й інші процедури оцінювання, автоматизувати які практично неможливо, наприклад ствердження про змістовність та інформативність повідомлень, які видаються програмним забезпеченням ІС, так само як і просто інформація про їх наявність або відсутність. Отже, деякі автоматичні засоби можуть виявитися корисними для оцінки якості програмного забезпечення, але набагато частіше потрібне тестування програмного забезпечення інформаційної системи експертом.

ОЦІНКА ЯКОСТІ ВРАХУВАННЯ ЧАСОВОГО ФАКТОРА

Характеристики предметної області, які відображаються в інформаційній системі, в більшості випадків з часом змінюються, що зумовлює необхідність як відображення таких змін в ІС, так і можливість отримання інформації про їх хронологію. Більшість інформаційних систем оперує тільки актуальною інформацією, яка потрібна у даний момент [1]. Сучасний стан розвитку інформаційних технологій обумовлює накопичення великої кількості різноманітних даних, котрі є вихідною інформацією для області видобування знань. Важливою характеристикою цих накопичених даних є різноманітні часові залежності інформаційних характеристик предметної області. Можливість врахування фактора часу в ІС сьогодні є дуже актуальною.

Загалом у проблематиці врахування часового фактора в інформаційних системах виділяють різноманітні задачі. До них можна віднести збереження, хронології станів предметної області, представлення й опрацювання різноманітних часових структур, нечітких часових параметрів та ін [13, 14].

Основою більшості ІС є база даних, і, відповідно, вирішення зазначених задач має в першу чергу відобразитись саме в ній. Отже, показники ІС для оцінки якості врахування часового фактора потрібно формувати в першу чергу для бази даних.

Згідно з наведеними вище підходами можна сформувати відповідне дерево показників (рис. 3).

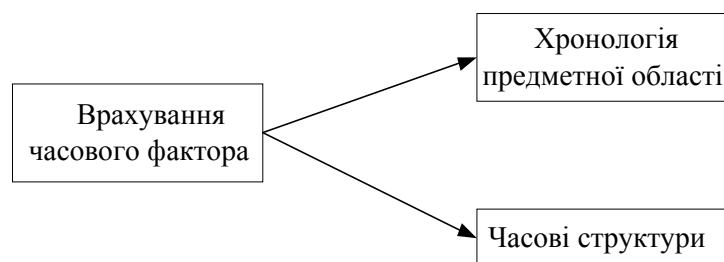


Рис. 3. Дерево характеристик якості врахування часового фактора

Загалом у врахуванні фактора часу можна виділити такі показники:

- а) хронологія предметної області. Цей показник характеризує міру того, наскільки повно в інформаційній системі відображається хронологія станів предметної області;
- б) часові структури. Це показник того, наскільки адекватно та повно відображені різноманітні інформаційні характеристики предметної області відносно різних часових структур (наприклад – дні тижня, пори року чи доби).

Наведені характеристики дають змогу оцінити міру того, наскільки якісно врахований часовий фактор в інформаційній системі. Оцінити виділені показники та розробити деякі числові міри для них можна на прикладі інформаційної системи для музейних установ.

ОЦІНКА ЯКОСТІ ВРАХУВАННЯ ЧАСОВОГО ФАКТОРА НА ПРИКЛАДІ ІС МУЗЕЙНОЇ УСТАНОВИ

Для застосування введених вище критеріїв оцінимо їх за запропонованими вище методиками. Для ІС музейної установи фактор часу є дуже важливим параметром, що дозволяє визначити її (ІС) якісні та кількісні характеристики. Для зручності оцінки самих критеріїв дані подано у вигляді таблиці:

Шифр характеристики	Опис	Кореляція з якістю	Потенційна вигода	Можливість числового подання
ФЧ-1	“хронологія предметної області”	МЗ	4	АА+КС
ФЧ-2	“часові структури”	3	5	АА+КС

Критерій ФЧ-2 дещо важливіший ніж ФЧ-1 саме для музейних установ, оскільки інформація про музейні експонати, як правило, містить різноманітні часові виміри, які рідко зустрічаються в стандартних засобах побудови ІС (наприклад, обмеження часових форматів СКБД).

Проведемо обчислення числових значень для виділених характеристик.

Для визначення числової метрики фактора ФЧ-1 можна скористатись таким виразом:

$$M_{tf1} = C_{infz}/C_{infn},$$

де C_{infz} – кількість інформаційних показників предметної області, історію яких збережено в базі даних ІС; C_{infn} – кількість інформаційних показників предметної області, які потенційно можуть змінювати своє значення протягом життєвого циклу ІС.

Як бачимо, виходячи з визначення, ці характеристики набуватимуть значення на проміжку від 0 до 1 ($0 \leq M_{tf1} \leq 1$).

Для числового подання фактора ФЧ-2 можна скористатись таким виразом:

$$M_{tf2} = C_z/C_n,$$

де C_{sz} – кількість часових інформаційних вимірів (варіантів поділу часової осі) предметної області, опрацювання яких забезпечено в базі даних ІС; C_{sn} – кількість часових інформаційних вимірів предметної області.

Аналогічно до попередньої характеристики справедливим буде $0 \leq M_{tf2} \leq 1$.

За запропонованими вище способами обчислення кількісних характеристик було отримано такі оціночні значення:

$$M_{tf1} = 476/1128 = 0.42; \quad M_{tf2} = 3/5 = 0.6.$$

ВИСНОВОК

У результаті проведення досліджень в даній роботі було отримано такі результати:

- а) сформовано формальний підхід до визначення показників якості інформаційних систем;
- б) запропоновано шкалу оцінки кореляції характеристик з загальною якістю ІС та можливістю визначення числових метрик;
- в) використано цей підхід для формування показників якості врахування часового фактора в ІС;
- г) на прикладі ІС музейної установи здійснено оцінку як самих сформованих характеристик, так і запропоновано алгоритми задання числових метрик цих характеристик.

1. Слисенко А.О. До проблеми аналізу логіко-алгоритмічної структури систем представлення й обробки знань // Питання кібернетики. – М.: НС по проблемі “Кібернетика”, 1989. – С. 85–97. 2. Chen Z. and Suen 3. Measuring the Complexity of Rule-Based Expert Systems // Expert Systems with Applications. – 1994. – Vol. 7, № 4. – P. 467–481. 3. Guida G. and Mauri G. Evaluating Performance and Quality of Knowledge-Based Systems: Foundation and methodology // IEEE Trans. Knowledge and Data Eng. – 1993. – Vol. 5, № 2. – P. 204–244. 4. Kaisler S. Expert System Metrics // Proc. of the IEEE Intern. Conf. on Systems, Man and Cybernetics. – 1986. – Vol. 1. – P. 114–120. 5. Кожевнікова Г.П. До побудови метричного апарату оцінки якості систем з базами знань // Пр. I-ї Міжнар. наук.-практ. конф. УкрПРОГ’98. – 1998. – С. 351–357. 6. Бабенко Л.П. Повторне використання в програмній інженерії: проблеми і підходи // Пр. I-ї Міжнар. наук.-практ. конф. Укрпрог’98. – 1998. – С. 129–136. 7. Кожевнікова Г.П. Структури даних і проектування ефективного обчислювального середовища. – Львів: Вища шк., 1986. – 176 с. 8. Бозм Б., Браун Дж., Каснар Х. и др. Характеристики качества программного обеспечения. – М.: Мир, 1981. – 206 с. 9. Захарова О.В. Функциональные зависимости как один из аспектов проверки полноты информационной модели // Инженерия и инструментальные средства программирования: Тез. докл. – К., 1–4 июня 1992; – К.: Ин-т программных систем АН Украины, 1993. – 116 с.

10. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 448 с. 11. Козаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с. 12. Критерии оценки и выбора, <http://www.citforum.ru/database>. 13. Ralph Kimball. The importance of the time dimension in data marts and data warehouses // DBMS – July 1997 <http://www.dbmsmag.com/9707d05.html>. 14. Андрей Прохоров, Informix Временной ряд как объект хранения в СУБД // CIT Forum <http://www.citforum.ru/seminars/cbd2001/day25informix.shtml>.

УДК 683.1

А.Я. Вовчина

Національний університет “Львівська політехніка”,
кафедра “Інформаційні системи та мережі”

ВИДІЛЕННЯ ЧАСОВИХ ІНТЕРВАЛІВ ПРИ ПОБУДОВІ КАЛЕНДАРНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

© Вовчина А.Я., 2006

In article approaches to division of a timebase into intervals structure of representation of time space in relational databases with formation optimum behind some attributes are considered. The primary goals of construction of time structures are allocated. The suggested campaign for definition of time dependences by division of time lines of values behind some criteria.

Розглянуто підходи до поділу часової осі на інтервали з формуванням оптимальної за деякими ознаками структури подання часового універсуму в реляційних базах даних. Виділено основні задачі побудови часових структур. Запропоновано підхід для визначення часових залежностей шляхом розбиття часового ряду значень за певними критеріями.

ВСТУП. ПОДАННЯ ЧАСУ В ІС

У більшості сучасних програмних засобів, зокрема СКБД, часові характеристики предметної області опрацьовуються з використанням загальноприйнятого структурування часу. Так, для подання деякої часової мітки на часовій осі, як правило, використовується структура типу <число:місяць:рік>, яка може бути доповнена структурою <година:хвилина:секунда>. Але таке загальноприйняте відображення деякої часової мітки не завжди є прийнятним при відображенні інформації про деяку предметну область в інформаційній системі її бази даних. Наприклад, в системі обліку робочого часу присутні такі часові виміри, як зміна, робочі /не робочі дні; в музейній справі – ера, епоха. Відображення та опрацювання таких часових інтервалів в реляційних базах даних породжує деякі проблеми, пов'язані з природою реляційної моделі даних. Також слід зазначити, що інформація, накопичена в