

В. Г. Пуятін

Інститут проблем реєстрації інформації НАН України
вул. Шпака, 2, 03113 Київ, Україна

Методологічні аспекти вибору компонентів корпоративної інформаційно-аналітичної системи з урахуванням критерію живучості

Розглянуто підходи щодо створення та вибору компонентів корпоративної інформаційно-аналітичної системи з урахуванням багатьох вимог і критерію живучості.

Ключові слова: *вибір, живучість, компонент, методика, корпоративна система, спосіб.*

1. Підхід щодо створення КІАС підвищеної живучості

Вирішальним фактором успішного розвитку сучасної організації (корпорації) є наявність у неї власної захищеної корпоративної інформаційно-аналітичної системи (КІАС), що дозволяє експертам, аналітикам і посадовим особам швидко аналізувати великі обсяги даних (інформації). КІАС — це особливий клас корпоративних інформаційних систем, призначених для аналітичної обробки великих обсягів різного роду даних, а не для автоматизації повсякденної діяльності організації [1]. КІАС об'єднують, аналізують і зберігають як єдине ціле інформацію, що отримується як з облікових баз даних (БД) організації, так і із зовнішніх джерел. При цьому під КІАС розуміється сукупність всієї різномірної інформації, що використовується в роботі організації, і комплексу програмно-технічних, методичних і організаційних компонентів, які забезпечують створення, обробку, передачу і прийом цієї інформації.

У загальному випадку до складу КІАС входять [2, 3]:

1) підсистема збору та зберігання корпоративних даних, що вирішує задачі збору та фільтрації даних, накопичення та індексування інформації, і забезпечує можливість використання інформації в аналітичних цілях, а також для підтримки прийняття рішень на різних рівнях управління;

2) підсистема доступу до даних, аналізу та корпоративної звітності, що включає модулі, які забезпечують доступ до даних і захист конфіденційної інформації, інструменти з підтримки прийняття рішень і оперативного аналізу інформації, засоби корпоративної звітності та візуалізації даних. В основі підсистеми лежить концепція сховищ даних (СД). СД, які входять до складу сучасних КІАС, забезпечують перетворення великих обсягів сильно деталізованих даних в узагальнену

достовірну інформацію, що придатна для прийняття обґрунтованих управлінських рішень. На відміну від звичайних баз даних СД містять оброблене, упорядковане представлення даних. СД є свого роду складальним конвеєром з підготовки інформації в інтегрованому, несуперечливому вигляді для підтримки прийняття управлінських рішень [3].

До сучасних КІАС підвищеної живучості висувається ряд основних вимог [1–5].

Адаптованість — можливість налаштування на мінливі потреби підрозділів організації, включаючи філії (відділення);

Адекватність — система повинна відповідати функціональним завданням, для вирішення яких вона створюється.

Живучість — КІАС підвищеної живучості повинна: 1) адаптуватися в мінливих умовах функціонування, протистояти несприятливим впливам і досягати мети функціонування за рахунок зміни поведінки та структури; 2) підтримувати найважливіші функції під час атак і збоїв, а також відновлювати всі функції в короткий термін; 3) виконувати встановлений обсяг функцій в умовах впливів зовнішнього середовища і відмов компонентів системи в заданих межах. В якості «несприятливих впливів» розглядаються можливі відмови, збої й порушення в роботі апаратного (АЗ) та програмного забезпечення (ПЗ), різноманітні атаки на систему і т.д., причому важлива не природа впливу, а його наслідки.

Масштабованість — система повинна зберегти адекватність при розвитку організаційної структури і зростанні інформаційного навантаження, не вимагаючи при цьому серйозної зміни архітектури системи. Масштабованість — одна з основних умов використання системи. При сучасному розвитку ПЗ та АЗ масштабність КІАС практично не обмежена при наданні відповідних технічних засобів. КІАС для забезпечення збереження вкладених у розробку і розвиток системи засобів повинна володіти високим ступенем масштабності при мінімальних часових і фінансових витратах за наступними напрямками: додавання нових автоматизованих робочих місць (АРМ) користувачів; розширення прикладних функцій; модернізація ПЗ; нарощування обсягу збережених даних; нарощування обчислювальних потужностей; збільшення швидкості обміну даними.

Модульність — КІАС підвищеної живучості повинна бути спроектована за модульним принципом.

Розширюваність — КІАС повинна поетапно розвиватися, дозволяючи включати і модифікувати морально застарілі, додавати нові компоненти. Розширюваність — здатність легко додавати нові функції до існуючих служб без зміни основних програм або без перевизначення основної архітектури; можливість розширення інформаційних ресурсів системи; можливість функціонального розвитку системи відповідно до потреб користувачів.

Збереження інвестицій — при модифікації системи слід максимально використовувати раніше придбане та встановлене обладнання та наявні компоненти КІАС підвищеної живучості.

Уніфікація — КІАС підвищеної живучості повинна мати розвинені технологічні засоби інтеграції з іншими прикладними системами і БД. При створенні системи необхідно використовувати принципи організації даних, що дозволяють при

технологічному і семантичному описі об'єктів застосовувати існуючі, що змінюються і знову вводяться, класифікатори та довідники.

Вимоги до технічного забезпечення — технічні засоби (ТЗ) і устаткування, що використовуються в КІАС, повинні відповідати задачам, які вирішуються, бути уніфікованими й надійними в роботі. Серверне обладнання КІАС повинно бути реалізоване у промисловому виконанні на високонадійній масштабованій платформі, з резервуванням найбільш відповідальних компонентів.

Традиційні підходи [2–4] до побудови КІАС виходять з того, що на початку проекту важко сказати, що повинне існувати в СД, і які аналітичні завдання будуть вирішуватися кінцевими користувачами. Наприклад, методологія Oracle DWM FT (Datawarehouse Method Fast Track — метод створення сховищ «високошвидкісна траса») виходить з припущення, що розробники протягом практично всього життєвого циклу КІАС будуть займатися визначенням і аналізом вимог до сховища даних. Заснована на DSDM (Dynamic System Development Method — метод розробки динамічних систем) методологія реалізує підхід RAD (Rapid Application Development — швидка розробка додатків).

КІАС зазвичай створюється поетапно. Схему її створення та експлуатації можна уявити наступним чином: вимоги замовника – задум розробника – проектування – впровадження і обслуговування – аналіз – модернізація. На етапі проектування системи закладається забезпечення всіх перерахованих вище вимог. Створення КІАС підвищеної живучості, які реально відповідають цілям і функціональним задачам організації, являє собою досить складний ітераційний процес, що включає етапи формування концепції, проектування, розробки, впровадження та супроводу. Сам характер цього процесу вимагає попередньої розробки досить жорсткої фіксованої технологічної схеми, яка представляє собою послідовність робіт і завдань, що виконуються певними виконавцями. Технологічна схема являє собою послідовність робіт і завдань, що виконуються певними виконавцями. Вона лежить в основі загальної методології створення КІАС підвищеної живучості, яка містить склад і послідовність робіт і завдань, склад рольових функцій і породжуваних артефактів (документів, моделей, схем тощо).

Методика створення таких КІАС охоплює наступні види діяльності: 1) збір, аналіз і деталізація вимог до системи, визначення пріоритетів реалізації цих вимог і постановка завдань з їхньої реалізації, визначення вимог до архітектури, надійності та захисту від несанкціонованого доступу і визначення складу даних; 2) розробка проектних рішень з усіх аспектів побудови КІАС, визначення складу джерел інформації, способів передачі та очищення даних, складу додатків організації доступу до даних, проектування архітектури, проектування БД; 3) розробка аналітичних програм, вибір і налаштування інструментальних засобів збору, перетворення й очищення даних і організації доступу користувачів до даних, розробка метаданих, тестування, розробка документації користувачів.

2. Підхід до створення потужних сучасних КІАС

Створення потужних сучасних КІАС на сьогоднішній день є одним із найбільш складних видів виробничої діяльності. Основною причиною цього є краща якість, властива процесу створення ПЗ. Це унікальна якість — гнучкість створен-

ня ПЗ: практично все, що завгодно, можна запрограмувати величезною кількістю способів. Крім того, стрімкий розвиток ІТ-індустрії, вихід все більше нових платформ, операційних систем (ОС), середовищ і технологій постійно збільшує цю множину, збільшуючи складність планування, моніторингу та управління розробкою ПЗ. Це дозволяє зробити висновок, що розробка та впровадження нових методико-технологічних підходів для ефективного створення програмних систем залишається вельми і вельми актуальним завданням.

Метою дослідження даної роботи є процес створення не будь-якого ПЗ, а тільки його підмножини — КІАС [6]. На даний час КІАС є одним з найбільш затребуваних класів ПЗ на ринку замовленого ПЗ, це пов'язано з глобальним процесом автоматизації інформаційної діяльності цілих галузей. При цьому, крім загальних проблем організації та управління їхнім створенням їм притаманні особливості, вимоги та проблеми більш конкретного характеру, що впливають як із специфіки функціонального призначення КІАС, так і умов, у яких створюються і використовуються КІАС: 1) вимоги до системи часто не є легко визначеними і/або добре відомими, задачі погано піддаються формалізації. Вимоги можуть змінюватися в циклі розробки КІАС; 2) необхідність демонстрації прототипу для визначення вимог або перевірки концепції; 3) особливості предметної області краще відомі користувачам, ніж розробникам КІАС; 4) проект може мати тип системної інтеграції (об'єднувати декілька існуючих систем); 5) проект може бути розширенням існуючої системи; 6) очікується тривала експлуатація системи в організації; 7) система повинна мати високий ступінь надійності (живучості); 8) можливо, система буде мінятися на етапі супроводу; 9) є істотні обмеження ресурсів (час, гроші, персонал).

Виходячи з цього в [7] зроблено висновок, що для оптимізації процесу створення КІАС необхідний новий комплексний методико-технологічний підхід, який враховує методичні, технічні та організаційні аспекти, перелічені вище. Тобто, даний підхід повинен включати весь необхідний спектр інструментів, які дозволяють ефективно і швидко створити реалізацію КІАС, при цьому працюючи по оптимальній моделі життєвого циклу проекту.

Дослідження [6, 7] включали аналіз існуючих методик (моделей) з точки зору ефективності при перелічених особливостях розробки КІАС. Аналіз даних моделей з урахуванням вищеперерахованих вимог показав, що найбільш ефективними в даному випадку є прототипування, швидка розробка (RAD) і інкрементна модель, але і вони мають ряд недоліків. Таким чином, для підвищення ефективності процесу розробки, що володіє перерахованими вище особливостями, необхідно створити інтегральну модель, яка поєднує в собі кращі особливості прототипування, RAD і інкрементної моделі, і при цьому виключає їхні недоліки. За основу взято модель RAD, яка задовольняє найбільшому числу особливостей процесу. Істотною рисою підходу буде те, що в якості інструментального середовища, яке використовується RAD, буде виступати технологічна платформа (ТП), що реалізує процес розробки КІАС у вигляді візуального налаштування спеціалізованих компонент.

При такому підході на кожному кроці розробки КІАС існує працездатна версія системи. Таким чином, розробка може йти за моделлю еволюційного прототи-

пування, за винятком того, що прототип у даному випадку буде не макетом результуючої системи, а повноцінною версією, наближенням до того, що треба реалізувати. Це дозволяє скористатись усіма плюсами прототипування без його недоліків (у класичній моделі прототипування окремо розробляється макет, а потім, коли всі вимоги зрозумілі, починається розробка повноцінної системи). Фактично, стирається грань між прототипуванням, розробкою та використанням програми. Цей підхід легко лягає в інкрементну модель, дозволяючи поступово нарощувати міць системи, яка розробляється (за винятком того, що інкремент-системи розробляються не за каскадною моделлю, а за адаптованою моделлю RAD-прототипування).

Відповідно до обраної моделі життєвого циклу, методика пропонує наступну послідовність дій: 1) аналіз вимог до системи; 2) формалізація предметної області; 3) розробка структури БД; 4) еволюційна розробка клієнтських місць КІАС в інструментальному середовищі технологічної платформи; 5) інтеграція ретроспективних джерел даних; 6) пробна експлуатація; 7) розгортання системи; 8) супроводження. Вимоги, що важко визначаються та змінюються в процесі реалізації проекту, враховуються циклічним характером етапу розробки, причому витрати, що пов'язані зі зміною вимог, мінімізуються за рахунок використання коштів, наданих технологічною платформою.

Проблеми, наведені вище, вирішуються за рахунок активного залучення користувачів на етапі розробки системи. Для вирішення завдань інтеграції та імпорту даних з існуючих систем у методиці передбачено окремий етап. За рахунок використання готових компонентів ТП і вбудованих механізмів управління пам'яттю підвищується надійність (живучість) системи, забезпечується контроль за технічними деталями розробки.

Використання ТП також дозволяє вносити зміни в систему на етапі супроводу та ефективно проводити розробку в умовах обмежень ресурсів. Далі розглянуто особливості розробленої ТП, яка враховує як вищенаведену інтегральну методу, так і особливості реалізації КІАС, викладені раніше. Основною складністю стала необхідність синтезу максимальної уніфікації та технологічного спрощення процесу розробки, який може забезпечити тільки візуальний підхід, із збереженням гнучкості і функціональних можливостей, які забезпечують класичні середовища розробки. Ключовими аспектами ТП з'явилися: 1) повністю візуальний підхід до розробки клієнтських місць (причому як для Windows, так і для веб-реалізації), виключення процесів кодування і компілювання; 2) зберігання опису інтерфейсного та функціонального клієнтських місць реалізованих КІАС на сервері БД; 3) універсальний клієнт для створення і доступу до КІАС.

На початковому етапі проектування архітектури ТП були виділені [7] логічні сутності, що складають інтерфейс будь-якої прикладної КІАС (компоненти КІАС): запити, діаграми, аналітичні звіти, форми для користувацького введення і т.д. Організуючи ці компоненти в ієрархічному порядку можна побудувати інтерфейс робочого місця КІАС. Для забезпечення гнучкості при проектуванні прикладної КІАС, було вирішено не кодувати жорстко цю ієрархію для кожної конкретної задачі, а зберігати на сервері БД разом з призначеними для користувача даними та описом цих компонентів.

Таким чином, реалізувавши застосування, здатне за описами компонентів будувати користувацький інтерфейс, отримано універсальний клієнт для доступу до множини прикладних КІАС, заснованих на цій технології. Тобто відсутня необхідність розробки спеціалізованого ПЗ для кожної з прикладних КІАС.

Далі, створивши для компонентів КІАС візуальні редактори, отримана можливість вести процес розробки КІАС без використання програмування, що значно знижує трудовитрати, підвищує швидкість розробки кінцевих систем, збільшує ступінь контролю за процесом, збільшує надійність реалізації.

Програмний комплекс складається з наступних основних частин: 1) сховища даних під управлінням промислової СУБД; 2) інструментального середовища конструювання КІАС, яка є також універсальним клієнтом доступу до КІАС в ОС Windows; 3) сервера додатків для публікації КІАС в Інтернеті з підтримкою ГІС-функцій.

Усі функціональні елементи КІАС зберігаються на сервері БД і доступні користувачам системи або за допомогою Windows-клієнта КІАС (з можливістю редагування елементів), або через веб-браузер (перегляд і редагування даних, без редагування елементів КІАС). Таким чином, елементи КІАС є надбудовою над даними предметної області, що дозволяє інтегрувати створювані системи з уже існуючими БД. При розробці візуального середовища конструювання КІАС величезна увага приділялася тому, щоб уникнути недоліків інших подібних засобів — чи все-таки присутня мова програмування, або створення інтерфейсу обмежене налаштованим набором шаблонів. Таким чином, відмова від класичного програмування повинна бути обґрунтована необмеженими можливостями середовища візуальної розробки.

Щоб забезпечити необхідну потужність при створенні користувацьких форм, була створена велика кількість візуальних компонентів управління даними, з широкими вбудованими можливостями (пошуку, фільтрації, бізнес-правил, і т.п.) Щоб повністю відмовитися від програмування, крім візуальних компонентів, великі зусилля прикладалися до розробки так званих «дій» — не візуальних компонентів, що реалізують обробку дій користувача КІАС. Крім того, так як КІАС працює, в першу чергу, з даними, то дані є і тим, що управляє поведінкою всіх компонент форми і забезпечує їхню логічну та функціональну ув'язку за допомогою передачі параметрів. Таким чином, компонуючи візуальні елементи призначеного для користувача інтерфейсу, дії, і зв'язуючи між собою елементи КІАС, виявилось можливим будувати робочі місця прикладної системи практично будь-якого ступеня складності. Дуже важливим моментом є те, що візуальне середовище редагування забезпечує єдино обране створення як Windows, так і веб-компонент.

Сервер застосувань, реалізований як служба Windows, є самостійним HTTP-сервером, що не вимагає для своєї роботи зовнішніх веб-серверів, таких як Internet Information Server або Apache. Крім протоколу HTTP, сервер застосувань підтримує закритий протокол передачі даних за технологією SSL, що дозволяє створювати захищені КІАС.

Технологічна платформа володіє власними засобами для організації роботи з просторовими даними. Усі картографічні і фактографічні дані КІАС зберігаються в єдиній БД. Для тих випадків, коли вбудованої функціональності з якої-небудь

причини не вистачає, у системі передбачена можливість вбудовування зовнішніх модулів розширення на основі технології COM (Component Object Model), що є основним засобом взаємодії між додатками в ОС Windows. ТП повністю підтримує стандарти COM для забезпечення взаємодії з іншими додатками. Система може виступати як сервер автоматизації, вбудовуючись у зовнішню програму. Для того щоб зовнішній додаток міг виступати безпосередньо в ролі контролера автоматизації, необхідно, щоб він підтримував технологію COM. Якщо ж цей додаток не підтримує COM, автоматизація КІАС можлива за допомогою виклику функцій динамічної бібліотеки-контролера, що трансліює зовнішні виклики у виклики COM-сервера.

Описаний вище підхід дає також ряд додаткових переваг. 1. Можливість одностайної побудови КІАС для Windows і Web. Реалізувавши універсальний клієнт для доступу до КІАС для Windows і Web, і створивши ієрархічну модель взаємодіючих компонентів КІАС, можна автоматично отримати Windows і веб-реалізації прикладної КІАС. 2. При зміні користувачем якого-небудь елемента КІАС ці зміни стають доступні для всіх клієнтів КІАС (Windows і Web). Причому, для отримання зміненого інтерфейсу не треба буде ні оновлювати програму-клієнта, ні навіть перезапускати її, достатньо буде лише знову відкрити змінений елемент КІАС. Таким чином, знімається проблема оновлення клієнтських місць. 3. Користувач-непрограміст може сам створювати КІАС, конструюючи її із зумовлених елементів, або доопрацьовуючи вже готову. 4. Системи, побудовані за допомогою ТП, більш надійні за системи, створені традиційним способом, за рахунок того, що використовуються багаторазово перевірені спеціалізовані компоненти. Є вбудовані механізми управління ресурсами і пам'яттю. 5. Поліпшується супровід системи, так як у випадку виявлення та виправлення помилки у стандартному компоненті ТП, ця помилка автоматично усунеться в усіх проектах, реалізованих на цій платформі. 6. Прискорюється термін розробки КІАС за рахунок використання готових рішень. 7. КІАС, побудовані за допомогою ТП, мають уніфікований вигляд і поведінку, що полегшує роботу в них користувачів. Попрацювавши з однією системою, реалізованою на ТП, користувач відчуває себе «як вдома», працюючи з будь-якими іншими системами, реалізованими на ній. 8. При використанні веб-версії КІАС в якості клієнтів системи виступають звичайні інтернет-браузери, ніякого додаткового програмного забезпечення на робочі місця встановлювати не потрібно.

Таким чином, технологічна платформа являє собою інтегрований методико-технологічний інструмент, що дозволяє максимально ефективно і швидко створювати потужні КІАС підвищеної живучості.

3. Спосіб вибору компонентів КІАС підвищеної живучості

До складу КІАС входять наступні компоненти [1–4]: функціональні підсистеми; забезпечувальні підсистеми; персонал. Функціональною підсистемою КІАС є сукупність функціональних задач (комплексів функціональних задач) з високим ступенем інформаційних обмінів між ними. Функціональна задача — певний процес обробки інформації з чітко визначеною множиною вхідної і вихідної інформації. До складу забезпечувальних підсистем КІАС входять організаційне, право-

ве, технічне, математичне, програмне, інформаційне, лінгвістичне та технологічне забезпечення.

Програмною системою КІАС є об'єднання функціональних задач (комплексу функціональних задач) певної функціональної підсистеми, фрагмента інтегрованої БД та інших компонентів системи з метою автоматизації певного виду діяльності організації.

До складу персоналу КІАС входять: технічні фахівці — адміністратори БД, прикладні програмісти, фахівці з супроводу ПС; кінцеві користувачі, які є працівниками організації; кінцеві користувачі, які уклали договір на доступ до інформаційних ресурсів системи; кінцеві користувачі, які звертаються до загальнодоступних інформаційних ресурсів КІАС через сайт організації.

Таким чином, компонентами КІАС у загальному випадку є: технічне та програмне забезпечення КІАС (мають характеристики, що дозволяють підтримувати всі етапи ЖЦ системи); інструментальні середовища, що використовуються при розробці інформаційного та програмного забезпечення КІАС; засоби підтримки захисту та безпеки даних (максимально забезпечується засобами СУБД, яка використовується); робочі станції користувачів; сервери додатків; сервери БД; кінцеві користувачі системи; адміністратори БД і прикладні програмісти; персонал КІАС. Для повноцінної роботи системи або збереження мінімального набору критично важливих функцій проєктована КІАС повинна володіти цілком певним (заданим) запасом стійкості до зовнішніх впливів дестабілізуючого середовища. При цьому, порушення цілісності системи на тлі зниження активності її складових компонентів тягне за собою дезорганізацію управління, одночасне зниження активності компонентів і їхньої живучості — втрату гнучкості, а зниження живучості та порушення цілісності системи — втрату найважливіших функцій [1–4].

У процесі проєктування КІАС також необхідно враховувати ряд параметрів живучості та надійності компонент (елементів) системи. Це дозволить виявити вузькі місця проєктованої системи, прогнозувати можливі збитки від руйнувань і надалі зміцнити або модернізувати систему, зводячи до мінімуму можливі втрати, вибрати оптимальні параметри і будувати раціональні стратегії управління.

Слід зазначити, що сучасні КІАС припускають включення у свою структуру ефективних систем моніторингу, що відстежують параметри стану системи для забезпечення контролю і безпеки. Також моніторинг передбачає виявлення різних неполадок чи погроз вірусних атак. Суттєва частка атак націлена на максимальне споживання ресурсів, що надаються, з метою значного погіршення або припинення надання послуг користувачам.

3.1. Забезпечення живучості КІАС

Для розробника КІАС украй важко орієнтуватися в системі документів нормативної бази відносно питання проєктування компонентів КІАС, стійких до аварійних дій. Такий стан нерідко призводить до утруднень при проходженні експертизи, з її неоднозначними багатоваріантними тлумаченнями визначень і понять [5–7]. Необхідний корінний перегляд методології створення КІАС підвищеної живучості. Розробникові потрібно забезпечити необхідний рівень надійності компонент у рамках КІАС. Надійність при цьому розуміється, як властивість систе-

ми виконувати своє функціональне призначення з необхідною якістю впродовж передбаченого терміну експлуатації, і оцінюється вірогідністю безвідмовної роботи.

При цьому виникають деякі труднощі. Так, при розробці КІАС виконується по-компонентний (по-елементний) розрахунок, тобто тим самим забезпечується необхідна надійність (вірогідність безвідмовності) кожного окремого компонента (елемента). Така по-компонентна (по-елементна) перевірка називається методом найслабкішого елемента. Оскільки визначити надійність усієї КІАС не представляється можливим з причини крайньої труднощі, то надійність усієї системи трактується через надійність її окремих елементів. У результаті не можна дати відповідь про фактичну надійність розробленої системи. Розробку КІАС слід розглядати в контексті узагальненої проблеми живучості.

Живучість — здатність пошкодженої системи адаптуватися до нових ситуацій, що змінилися і, як правило, непередбачених, протистояти шкідливим впливам, виконуючи при цьому повністю або частково свою цільову функцію, за рахунок відповідної зміни структури і поведінки системи. Живучість розуміється як стійкість системи у запроектній (нештатній) ситуації. Залежно від міри складності організації і класу систем у рамках сучасної теорії катастроф, а також рівня аналізу, властивість живучості може проявлятися як складна інтегральна якість системи і, відповідно, кількісно оцінюватися показниками стійкості, надійності, адаптивності, відмовостійкості тощо. Властивість живучості забезпечується і безпосередньо взаємозв'язана з властивістю надійності. Живучість не можна гарантувати, якщо немає необхідного рівня вірогідності безвідмовної роботи частини системи, яка залишилася. Живучість забезпечується в комплексі всіх систем життєзабезпечення системи — інформаційного забезпечення, даних тощо.

Основним загально визнаним прийомом забезпечення надійності КІАС є резервування апаратури, програм і даних. Це вірно і «в малому», для окремих компонентів (пристроїв), коли до них вводять надлишкові модулі, і для КІАС у цілому, коли для забезпечення живучості системи створюють резервний центр. У більшості організацій КІАС знаходяться в постійному розвитку, тому впродовж всього ЖЦ системи слід співвідносити всі зміни з необхідністю забезпечення живучості.

Програми та дані динамічніші, ніж апаратура, і резервуватися вони можуть постійно, при кожній зміні, після завершення деякої логічно замкнутої групи змін або після закінчення певного інтервалу часу. Резервування програм і даних може виконуватися багатьма способами — за рахунок дзеркалювання дисків, резервного копіювання й відновлення, реплікації БД і т.п. Будемо використовувати для всіх перерахованих способів термін «тиражування». У наступному пункті зроблено спробу проаналізувати загальні властивості тиражування як засобу забезпечення живучості програм і даних.

3.2. Класи тиражування

Виділяються наступні класи тиражування: 1) симетричне/асиметричне. Тиражування називається симетричним, якщо всі сервери, що надають даний сервіс, можуть змінювати інформацію, яка їм належить, і передавати зміни інших серверів. В іншому випадку тиражування називається асиметричним; 2) синхронне/асинхронне. Тиражування називається синхронним, якщо зміна передається всім

екземплярам сервісу в рамках однієї розподіленої транзакції. В іншому випадку тиражування називається асинхронним; 3) таке, що здійснюється засобами сервісу, і зберігає інформацію зовнішніми засобами. Сучасні сервіси, такі як СУБД, надають всі перераховані можливості.

Розглянемо, які з них кращі. У відповідності з рекомендованою політикою підвищення доступності, слід віддати перевагу стандартним засобам тиражування, які вбудовані в сервіс. Асиметричне тиражування ідейно простіше симетричного (виключені конфлікти щодо змін), тому доцільно вибрати асиметрію. Найважчим є вибір між синхронним і асинхронним тиражуванням.

Синхронне тиражування ідейно простіше (немає періоду неузгодженості сервісів, коли транзакція успішно завершилася, але її результати ще не протиражировані; якщо в цей період трапиться відмова основного сервера, клієнт буде вважати, що замовлена ним операція пройшла нормально, проте на резервному сервері її результати будуть відсутні), але його реалізація може бути великоваговою і складною, хоча це внутрішня складність сервісу, невидима для додатків.

Асинхронне тиражування стійкіше до відмов у мережі, воно менше впливає на роботу основного сервісу. Чим надійніший зв'язок між серверами, залученими в процес тиражування, чим менший час, що відводиться на переключення з основного на резервний сервер, і чим жорсткіші вимоги до актуальності інформації, тим більш кращим виявляється синхронне тиражування. Правда, необхідно також брати до уваги особливості конкретних сервісів, апробованість реалізації синхронного тиражування, ступінь впливу тиражування на роботу користувачів.

З іншого боку, недоліки асинхронного тиражування можуть компенсуватися процедурними і програмними заходами, спрямованими на контроль цілісності інформації у розподіленій КІАС. Сервіси, що входять до складу КІАС, здатні забезпечити ведення і зберігання журналів транзакцій, за допомогою яких можна виявляти операції, загублені при перемиканні на резервний сервер. Навіть в умовах нестійкого зв'язку з віддаленими філіями організації, подібна перевірка у фоновому режимі займе не більше декількох годин, тому асинхронне тиражування може розглядатись як кандидат на використання практично в будь-якій КІАС. Асинхронне тиражування може вироблятися на сервер, що працює в режимі гарячого резерву, можливо, навіть який обслуговує частину призначених для користувача запитів, або на сервер, що працює в режимі теплового резерву, коли зміни періодично накочуються, але сам резервний сервер запитів не обслуговує.

Перевага теплового резервування у тому, що його можна реалізувати, здійснюючи менший вплив на основний сервер. Цей вплив взагалі може бути зведений до нуля, якщо асинхронне тиражування здійснюється шляхом передачі інкрементальних копій з основного сервера (резервне копіювання необхідно здійснювати в будь-якому випадку). Друга перевага теплового резерву полягає в тому, що його можна використовувати як частину імітаційного стенду і відпрацьовувати на ньому технічні та процедурні рішення в умовах, що максимально наближені до реальних. Основний недолік теплового резерву полягає у відносному великому часі включення, що може бути неприйнятним для «важких» серверів, таких як кластерна конфігурація сервера СУБД. Тут необхідне проведення вимірювань в умовах, близьких до реальних. Може виявитися, що в найпотрібніший момент термі-

новий перехід резерву в штатний режим неможливий. Враховуючи наведені міркування, слід у першу чергу розглядати можливість гарячого резервування, або ретельно контролювати використання теплого резерву та регулярно (не рідше 1 разу на тиждень) проводити пробні перемикання резерву в гарячий режим.

4. Можливий підхід до аналізу та вибору компонентів КІАС

Життєвий цикл КІАС включає стратегічне планування, аналіз, проектування, реалізацію, впровадження та експлуатацію. Кожний етап ЖЦ характеризується певними задачами і методами їхнього вирішення, вихідними даними, отриманими на попередньому етапі, і результатами. При аналізі компонент КІАС їх слід розглядати не локально, а у комплексі, що дозволяє реально охарактеризувати їхні переваги, недоліки та місце у загальному технологічному циклі створення КІАС підвищеної живучості.

У загальному випадку стратегія вибору компонент КІАС для конкретного застосування залежить від наступних чинників: 1) характеристик модельованої предметної області; 2) цілей, потреб і обмежень майбутнього проекту КІАС, включаючи кваліфікацію фахівців, які беруть участь у процесі проектування; 3) методології проектування, яка використовується.

Методологія проектування компонент КІАС підвищеної живучості визначається як сукупність трьох складових: 1) *покрокової процедури*, яка визначає послідовність технологічних операцій проектування; 2) *критеріїв і правил*, які використовуються для оцінки результатів виконання технологічних операцій; 3) *нотацій* (графічних і текстових засобів), що використовуються для опису проектованої системи.

На вибір компонент КІАС підвищеної живучості можуть істотно вплинути такі особливості методології проектування: орієнтація на створення унікального або типового проекту КІАС; ітераційний характер процесу проектування; можливість декомпозиції проекту КІАС на складові частини, що розробляються групами виконавців обмеженої чисельності з подальшою інтеграцією складових частин; жорстка дисципліна проектування і розробки при їхньому колективному характері; необхідність відчуження проекту КІАС від розробників і його подальшого централізованого супроводу; засоби проектування компонент КІАС, що використовуються.

Під засобами проектування КІАС (ЗП КІАС) будемо розуміти комплекс інструментальних засобів, що забезпечують у рамках обраної методології проектування підтримку повного ЖЦ. Традиційно під час обговорення проблеми вибору ЗП компонент КІАС підвищеної живучості велика увага приділялася особливостям реалізації тієї чи іншої методології аналізу предметної області (ER, IDEF0, IDEF1X, Gane/Sarson, Yordon, Barker та ін.) Безумовно, багатство образотворчих і описових засобів дає можливість на етапах стратегічного планування та аналізу побудувати найбільш повну та адекватну модель предметної області. З іншого боку, якщо говорити про кінцеві результати — БД і застосуваннях, то виявляється, що частина описів у них практично не відбивається, залишаючись суто декларативною (на виході ми у будь-якому випадку отримуємо опис БД у табличному представленні з мінімальним набором обмежень цілісності і здійснимий код зас-

тосувань, більшу частину яких складають екранні форми, не виведені безпосередньо з моделей предметної області).

Можна сформувати необхідну апаратну платформу з компонентів різних фірм-виробників, так само можна вибрати і скомплексувати різні компоненти КІАС підвищеної живучості. Оскільки складові частини проекту повинні бути інтегровані в єдиний продукт, має сенс розглядати не будь-які, а тільки зв'язані компоненти, які в принципі можуть бути орієнтовані — навіть всередині одного класу — на різні методології.

Виходячи з перерахованих вище міркувань, візьмемо як основні критерії вибору ЗП компонент КІАС наступні.

1. *Підтримка повного ЖЦ КІАС із забезпеченням еволюційності її розвитку.* Повний ЖЦ КІАС підвищеної живучості повинен підтримуватися «наскрізним» технологічним ланцюжком, що забезпечує рішення наступних завдань: 1) обстеження та отримання формалізованих знань про предметну область (послідовний і логічно зв'язаний перехід від формалізованого опису предметної області до її моделей), 2) декомпозиція проекту на складові частини та інтеграція складових частин; 3) проектування моделей додатків (логіки додатків і користувальницьких інтерфейсів), 4) прототипування додатків; 5) проектування БД; 6) колективна, територіально розподілена розробка програм з використанням різних інструментальних засобів (включаючи їхню інтеграцію, тестування і налагодження); 7) розробка розподілених БД; 8) розроблення проектної документації з урахуванням вимог проектних стандартів; 9) адаптація до різних системно-технічних платформ і СУБД; 10) тестування та випробування; 11) супровід, внесення змін і керування версіями і конфігурацією КІАС; 12) інтеграція з існуючими розробками (включаючи реінжиніринг додатків, конвертація БД); 13) адміністрування КІАС (оптимізація експлуатаційних характеристик); 14) управління розробкою і супроводом КІАС (планування, координація та контроль за ресурсами та ходом виконання робіт); 15) прогнозування та оцінка трудомісткості, строків і вартості розробки. Для існуючих КІАС повинен забезпечуватися плавний перехід із старого середовища експлуатації до нового з мінімальними переробками і підтримкою експлуатованих БД і додатків, впроваджених до початку робіт зі створення нової системи.

2. *Забезпечення цілісності проекту та контролю за його станом.* Дана вимога означає наявність єдиного технологічного середовища створення, супроводу і розвитку КІАС, а також цілісність бази проектних даних (сховища). Єдине технологічне середовище має забезпечуватися за рахунок використання єдиної CASE-системи для підтримки моделей КІАС, а також за рахунок наявності програмно-технологічних інтерфейсів між окремими інструментальними засобами, сертифікованих і підтримуваних фірмами-розробниками відповідних засобів.

3. *Незалежність від програмно-апаратної платформи і СУБД.* Вимога визначається неоднорідністю середовища функціонування КІАС. Така незалежність може мати дві складові: незалежність середовища розробки і незалежність середовища експлуатації додатків. Вона забезпечується за рахунок наявності сумісних версій ЗП компонент КІАС для різних платформ і драйверів відповідних мережевих протоколів, менеджерів транзакцій і СУБД.

Крім перерахованих основних критеріїв, попередній аналіз при виборі ЗП компонент КІАС повинен враховувати наступні аспекти.

1. *Можливість розробки додатків «клієнт-сервер» необхідної конфігурації.* Мається на увазі поєднання наявності розвинутого графічного середовища розробки додатків (багатовіконність, різноманітність стандартних графічних об'єктів, різноманітність шрифтів, які використовуються, і т.д.) з можливістю декомпозиції додатків на «клієнтську» частину, що реалізовує користувальницький екранний інтерфейс і «серверну» частину. При цьому повинна забезпечуватися можливість переміщення окремих компонентів додатка між «клієнтом» і «сервером» на найбільш підходящу платформу, що забезпечує максимальну ефективність функціонування програми в цілому, а також можливість використання сервера додатків (менеджера транзакцій).

2. *Відкрита архітектура і можливості експорту/імпорту.* Відкрита та загальнодоступна інформація щодо форматів даних і прикладних програмних інтерфейсів, які використовуються, повинна дозволяти інтегрувати інструментальні засоби третіх фірм і відносно безболісно переходити від однієї системи до іншої. Можливості експорту/імпорту означають, що специфікації, отримані на етапах аналізу, проектування і реалізації для однієї КІАС, можуть бути використані для проектування іншої КІАС. Повторне проектування і реалізація можуть бути забезпечені за допомогою засобів експорту/імпорту специфікацій у різні ЗП.

3. *Якість технічної підтримки, досвід успішного використання.* Мається на увазі наявність кваліфікованих консультантів, швидкість обслуговування користувачів, висока якість технічної підтримки і навчання продукту і методології його застосування для великих колективів розробників (наявність відомостей про практику використання системи, якість документації, укомплектованість прикладами та навчальними курсами, наявність прототипу проектів). Витрати на навчання новим технологіям значні, проте втрати від використання сучасних складних технологій ненавченими фахівцями можуть виявитися значно вищими. Крім того, фірми-постачальники інструментальних ЗП компонент КІАС повинні бути стійкими, оскільки технологія вибирається не на один рік, а також повинні забезпечувати хорошу підтримку (гаряча лінія, консультації, навчання, консалтинг), можливо, через дистриб'юторів.

4. *Простота використання.* Враховуються такі характеристики ЗП: доступність для користувача інтерфейсу; час, необхідний для навчання; простота інсталяції; якість документації.

5. *Використання загальноприйнятих, стандартних нотацій і угод.* Для того, щоб проект міг виконуватися різними колективами розробників, необхідне використання стандартних методів моделювання і стандартних нотацій, які повинні бути оформлені у вигляді нормативів до початку процесу проектування. Недотримання цієї вимоги ставить розробників у залежність від фірми-виробника даного компонента, робить скрутним формальний контроль коректності нотацій, які використовуються, і знижує можливості залучення додаткових колективів розробників, оскільки число фахівців, знайомих з даним методом (нотацією) може бути обмеженим.

В ідеальному випадку остаточний вибір може бути зроблений за результатами тестування відповідно до заданого плану, яке повинно включати імітацію проектування реальної БД і розробки додатків і складатися з наступних кроків:

1) *установка і конфігурація* (ясність і точність інструкцій з установки, наявність підказок у процесі установки, можливість установки по вибору і задання розрахованої на багато користувачів конфігурації);

2) *розробка концептуальної схеми БД* (зрозумілість і простота побудови, модифікації та документування різних елементів діаграм «сутність-зв'язок», відображення обмежень посилальної цілісності і бізнес-правил, управління режимом відображення);

3) *формування звіту про концептуальну схему* (список сутностей із визначеннями і атрибутами, включаючи вказівку ключів, список атрибутів, згрупованих за сутностями, список зв'язків між сутностями, можливість форматування звіту, складання звіту з виділеної частини схеми, передача звіту, наприклад, в інші програми (текстові процесори);

4) *розробка графічної схеми БД для конкретної СУБД з урахуванням специфічних для неї структур даних і обмежень* (вибір цільової СУБД і реалізація елементів схеми — введення та модифікація імен таблиць і стовпців, визначення типів даних, доменів, індексів, значень за замовчуванням і невизначених значень, порядку індексування, а також завдання обмежень посилальної цілісності та додаткових бізнес-правил, що характеризують предметну область, управління тригерами і збереженими процедурами);

5) *формування звіту про схему БД* (друк діаграми схеми, списку таблиць з відповідними стовпцями, первинними ключами, індексами і т.д., можливість форматування звіту, складання звіту з виділеної частини схеми, передачі звіту в інші програми);

6) *генерація схеми БД* (трансформація схеми БД в файл DDL в текстовому форматі або безпосередній інтерфейс з цільовою СУБД);

7) *розробка найпростішого додатку* (опис екранних форм, програмування або опис логіки програми та інтерфейсу з БД, завантаження БД тестовими даними та тестування додатка);

8) *супровід схем БД* (створення нових сутностей і атрибутів, зміна схеми БД, повторна генерація схеми, управління версіями, забезпечення збереження даних, синхронізація концептуальної схеми і самої БД);

9) *зворотне проектування* — реінжиніринг (повне і точне відновлення вихідної концептуальної схеми по файлах DDL або безпосередньо зі словника цільової СУБД).

У результаті виконаного аналізу може виявитися, що жодний доступний засіб не задовольняє в потрібній мірі всім основним критеріям і не покриває всі потреби проекту. У цьому випадку може застосовуватися набір засобів, що дозволяє побудувати на їхній базі єдине технологічне середовище.

5. Вибір прикладного ПЗ і апаратної платформи КІАС

Головним завданням КІАС є інформаційно-аналітична підтримка виробничих, адміністративних та управлінських процесів (далі — бізнес-процесів), що

формують продукцію або послуги організації (підприємства). Після того, як рішення про організацію бізнес-процесів в організації прийнято, важливим етапом є вибір прикладного програмного забезпечення (ППЗ), яке буде призване обслуговувати і автоматизувати бізнес на підприємстві. Багато компаній використовують наступний, у принципі цілком можливий варіант — вони стверджують: «Ми маємо в штаті програміста, і він може запрограмувати все від самого початку, до самого кінця базовою мовою C++ або Delphi».

Однак, як показує практика, основні недоліки «самопальних» систем з'ясовуються з часом уже на етапах їхньої експлуатації і ведуть до руйнівних наслідків, оскільки виправлення помилок вимагає великих капітало- і трудовкладень, а іноді виявляється необхідним зупиняти систему на невизначений термін, що тягне за собою фактичне загальмування бізнесу в ряді напрямків, які безпосередньо контролювалися за допомогою системи.

При виборі постачальника ППЗ важливим чинником є його фінансова стабільність, тому що фінансово нестабільний постачальник програмно-прикладної компоненти КІАС набагато гірше, ніж фінансово нестабільний клієнт.

Можна сформулювати ряд критеріїв, якими слід керуватися при підборі системи ПЗ: 1) система повинна бути саме системою, тобто зміна в одній її частини повинні автоматично змінити показники в інших її частинах (цю властивість системи прийнято називати інтегрованістю); 2) процедури в КІАС повинні бути дійсно автоматизовані; 3) система повинна забезпечувати реалізацію бізнес-процесів і процедур, які існують або повинні існувати (оптимальні для конкретної організації); 4) система повинна давати керівникові можливість отримувати оперативну інформацію в обсязі, достатньому для прийняття оперативних рішень; 5) система повинна бути легкою у навчанні і використанні (дружня), щоб пересічний працівник міг навчитися виконувати свої обов'язки з її допомогою за максимально коротким часом; 6) у системі повинна бути закладена можливість без допомоги програміста редагувати всі необхідні звіти і документи, змінювати їхню форму та створювати власні формати; 7) у системі повинні бути закладені процедури контролю, що зводять помилки до мінімуму; 8) система повинна давати можливість відстежити, хто і коли вніс зміни в тому чи іншому файлі, і який запис був до цих змін; 9) у системах середнього рівня і вищих, повинні бути присутніми надійні програми захисту даних і функції розподілу прав доступу.

Після того, як рішення про вибір ППЗ, на базі якого буде побудована КІАС підвищеної живучості, прийняте, необхідно прийняти рішення з апаратної платформи. Апаратну платформу КІАС слід вибирати, проаналізувавши і визначивши перспективи розвитку організації, беручи до уваги масштаб вибраного ППЗ і те, чи планується його нарощування в майбутньому, а також всілякі кількісні та якісні параметри, такі як кількість АРМ, їхні функції, обсяги інформації, спрямованість її потоків. Масштаби і потужність системи визначають рівень технічного рішення. А до рівня і конкретного наповнення КІАС вже можуть бути застосовані вартісні мірки. Доводиться шукати оптимальне рішення між нерозумною, з економічної точки зору, надмірністю і втратою капіталовкладень у програмно-технічну платформу.

6. Вибір технологічної архітектури КІАС

Технологічна архітектура КІАС — визначення технологічних параметрів (сукупність програмного і технічного забезпечення) КІАС. Основу КІАС складає комплекс взаємозалежних моделей з відповідною інформаційною підтримкою дослідження, експертні й інтелектуальні системи, що включають досвід вирішення задач управління і забезпечують участь колективу експертів у процесі вироблення раціональних рішень. У вузькому сенсі під архітектурою розуміється архітектура набору команд. Архітектура набору команд служить кордоном між АЗ, ПЗ і представляє ту частину системи, яку видно розробнику. У широкому сенсі архітектура охоплює поняття організації системи, що включає високорівневі аспекти розробки комп'ютерних та інформаційних складових (компонентів).

Стосовно до КІАС термін «архітектура» може бути визначений як розподіл функцій, що реалізуються системою, між її рівнями, точніше як визначення меж між цими рівнями. Архітектура КІАС передбачає багаторівневу організацію. У нашому розумінні архітектура КІАС це багаторівневе представлення КІАС та її компонентів, що розкриває зміст рівнів, способи міжрівневої взаємодії.

До складу архітектурно-технологічної схеми КІАС входять [7]: джерела інформації (зовнішні і внутрішні); централізоване сховище даних (включаючи підсистему інформаційного обміну та телекомунікаційну підсистему); підсистема прогнозно-аналітичних розрахунків (підсистема моніторингу; аналітична підсистема; підсистема моделювання і прогнозування; підсистема представлення даних; підсистема вироблення управлінських рішень за заданими критеріями); апаратно-технологічна система (засоби відображення інформації; сервери; підсистема резервування; засоби зв'язку); підсистема адміністрування та інформаційної безпеки.

Централізоване сховище даних КІАС призначене для накопичення та зберігання історичних даних за певними показниками територіальної структури корпорації.

Підсистема інформаційного обміну призначена для забезпечення інформаційної взаємодії; імпорту даних із встановлених форматів і різних СУБД до сховища даних і експорту даних у встановлені формати.

Підсистема прогнозно-аналітичних розрахунків призначається для проведення моніторингу, аналізу поточної ситуації, сценарного та цільового прогнозування розвитку корпорації в залежності від керуючих впливів.

Підсистема представлення даних призначена для відображення даних в табличній, графічній та картографічній формах.

Підсистема вироблення управлінських рішень за заданими критеріями призначена для підтримки вироблення управлінських рішень на основі комплексу заданих критеріїв.

Підсистема адміністрування та інформаційної безпеки призначена для забезпечення захисту елементів і даних від НСД та інформаційної безпеки.

Вибір апаратно-технологічної системи. Апаратно-технологічна система (АТС) — комплекс програмно-апаратних засобів, що забезпечують доставку відомчої інформації з різних джерел, консолідоване зберігання інформації, управління передачею даних, надійне резервування і відновлення даних, високошвидкісний обмін інформацією між вузлами підсистем КІАС і дисковими пристроями

зберігання даних, управління потоком транзакцій, ініційованих користувачами і компонентами самої системи в процесі роботи. АТС повинна включати в себе серверне обладнання, дискові масиви, окремі технологічні РС, сервери друку, зовнішні накопичувачі, принтери, сканери, ОС, СУБД та інші програмні середовища, що забезпечують працездатність ТЗ, програмну діагностику і керування технічними засобами.

Вибір програмного забезпечення КІАС. Програмне забезпечення, що реалізує функції КІАС, має включати в себе: ЗПЗ, ППЗ. Загальносистемне ПЗ повинне включати ліцензійні програмні продукти з наступним призначенням: ОС серверів; ОС робочих станцій; СУБД. ОС серверів повинна володіти функціональними можливостями для реалізації вимог інформаційного забезпечення КІАС. Вимоги до ОС робочих станцій повинні включати в себе: підтримку стеків протоколів TCP/IP; підтримку роботи 32-х розрядних додатків, підтримку роботи офісних додатків MS Office. Основними вимогами до СУБД є: конфігурування СУБД для максимального використання можливостей серверного обладнання з урахуванням можливих варіантів конфігурації (кластерної, багатопроцесорної і т.д.); конфігурування СУБД для оптимального управління БД великого (до 10 Тбайт) обсягу.

Основними вимогами до офісних додатків є: підтримка роботи користувачів з текстовими, табличними та графічними документами в середовищі встановленої ОС; можливість прийому і відправки повідомлень електронною поштою в середовищі встановленої ОС; підтримка роботи користувачів за допомогою інтернет-браузерів у середовищі встановленої ОС. Вимоги для серверної частини: Сервер БД; ОС сервера БД. Склад і конкретні версії ЗПЗ повинні бути визначені на стадії проектування КІАС з урахуванням наявних програмно-технічних засобів і особливостей ППЗ.

Вибір технічного забезпечення КІАС. КІАС не повинна вимагати для свого функціонування використання унікального технічного забезпечення (ТЗ). В якості апаратної платформи для КІАС повинні служити ТЗ одного з великих виробників сучасної обчислювальної техніки (ОТ), здатної забезпечити своєчасний і якісний супровід своїх продуктів, що випускає засоби ОТ, побудовані на основі сучасних і перспективних рішень, що мають широкий діапазоном обчислювальної потужності та працюють під управлінням ОС, що побудовані на основі стандарту відкритих систем.

Засоби технічного забезпечення КІАС повинні складатися з ТЗ сервера, ТЗ РС і локальної та зовнішньої інформаційної мережі. ТЗ КІАС повинне включати до свого складу: серверний комплекс, інтегрований в інформаційно-обчислювальну мережу користувачів КІАС; Робочі станції, які мають забезпечувати роботу користувачів на відповідних АРМ; пристрої для резервного копіювання і зберігання даних КІАС; засоби виведення даних на друк; засоби безперебійного живлення. Мінімальні вимоги до конфігурації сервера і робочої станції встановлюються на етапі розробки Технічного Завдання і залежать від конкретних рішень у рамках КІАС.

1. Додонов А.Г. Компьютерные информационно-аналитические системы. Толковый словарь. / А.Г. Додонов, Д.В. Ландэ, В.Г. Путятин. — К.: Наук. думка, 2011. — 384 с.

2. Додонов А.Г. Организация структуры Правительственной информационно-аналитической системы по вопросам чрезвычайных ситуаций // А.Г. Додонов, В.Г. Путятин, В.А. Валетчик // Электронное моделирование. — 2006. — Т. 28, № 3. — С. 61–82.
3. Додонов А.Г. Построение информационно-аналитической системы научно-исследовательского испытательного полигона // А.Г. Додонов, В.Г. Путятин, В.А. Валетчик // Управляющие системы и машины. — 2006. — № 4. — С. 3–14.
4. Галахов Илья. Проектирование корпоративной информационно-аналитической системы / Илья Галахов // Открытые системы. — 2003. — № 04. — С. 27–32.
5. Додонов А.Г. Введение в теорию живучести вычислительных систем / А.Г. Додонов, Е.С. Горбачик, М.Г. Кузнецова. — К.: Наук. думка, 1990. — 184 с.
6. Аракчеев Д.Б. Программно-инструментальные средства для разработки информационно-аналитических систем / Д.Б. Аракчеев // Геоинформатика. — 2004. — № 2. — С. 37–45.
7. Попов А.С. ИАС-КОНСТРУКТОР: технологическая платформа для разработки распределенных информационно-аналитических систем / А.С. Попов // Геоинформатика. — 2006. — № 2. — С. 44–53.

Надійшла до редакції 01.09.2015