

4. Пайлон Д., Питмен Н. UML 2.0 для програмістів. СПб.: Питер, 2012. 240 с. 5. Калянов Г.Н. CASE-технології. Консалтинг в автоматизації бізнес-процесів. М.: Горячая линия-Телеком, 2002. 320 с. 6. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя: пер. с англ. М.: МДК, 2000. 429 с. 7. Евланов М.В., Корнеева Е.В. Проблема автоматизации разработки и корректировки моделей бизнес-процессов промышленного предприятия // Материалы II Международной научно-практической конференции "Перспективные разработки науки и техники - '2006". Днепропетровск: Наука и образование, 2006. С. 64-66.

Надійшла до редколегії 15.01.2018

Євланов Максим Вікторович, д-р техн. наук, доцент, професор кафедри ІУС ХНУРЕ. Наукові інтереси: проблеми інтелектуалізації проектування і експлуатації інформаційних управляючих систем. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 70-21-451.

Корнєєва Євгенія Володимирівна, економіст, Дніпровський державний проектний інститут житлового та цивільного будівництва. Наукові інтереси: проблеми розробки інформаційних технологій моніторингу та управління бізнес-процесами підприємства. Адреса: Україна, 49000, м. Дніпро, вул. Січеславська набережна, буд. 29 А, тел. 056-370-30-56.

УДК 004.414.28

DOI: 10.30837/0135-1710.2018.175.074

О.Д. КАРАБИЦЬКА

МЕТОД ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ВІДКРИТОЮ АРХІТЕКТУРОЮ

Проаналізовано існуючі методи проектування інформаційної системи з відкритою архітектурою. Пропонується удосконалення моделі рішення задачі синтезу структури створеної інформаційної системи і методу вирішення цієї задачі. Наведені результати апробації дозволяють стверджувати, що запропоноване удосконалення скорочує обчислювальні витрати на реалізацію методу.

1. Постановка проблеми розвитку сучасних інформаційних систем

Процеси інформатизації в суспільстві базуються на ідеї інтеграції, яка проявляється в об'єднанні створених раніше і створюваних в даний час інформаційних систем (ІС). ІС визначають, як комплекс інформації, математичних методів і моделей, технічних, програмних, технологічних засобів і фахівців, призначений для обробки інформації та прийняття управлінських рішень. На даний момент виділяють ряд недоліків сучасних ІС [1,2]:

- чутливість системи - будь-яка невірна інформація або її відсутність веде до помилок в роботі системи і до високого ризику прийняття невірної рішення;
- дорожнеча рішення - крім вкладень на створення ІС, утримання і модифікація існуючої системи потребують постійних витрат;
- проблеми розуміння - відсутність у працівників компанії розуміння щодо складу функцій ІС, необхідних цій компанії;
- проблеми організації - відсутність формалізованої системи, що впливає на розвиток компанії;
- проблеми управління - відсутність правил прийняття рішень, відсутність способів контролю якості роботи, відсутність зрозумілого інструментарію прийняття управлінських рішень;
- проблеми автоматизації - відсутність програмних інструментів і кваліфікованих фахівців, здатних реалізувати специфічні технології, що використовуються або плануються до використання компанією, необхідність регулярної модернізації системи з метою повнішої відповідності ІС потребам підприємства;
- технічні збої і поломки програмного забезпечення (ПЗ) - технічні збої і несправності ПЗ, потреба в регулярному оновленні версій ПЗ, необхідність навчання нових користувачів ПЗ;
- проблеми розвитку ІС - необхідність серйозної модифікації системи при будь-якій, навіть мінімальній, зміні процесу прийняття управлінського рішення.

Найважливішою з представлених проблем є проблема розвитку системи. Спроби вирішення даної проблеми різними ІТ-компаніями визначили необхідність створення інформаційних систем з відкритою архітектурою (ВІС). Під ВІС слід розуміти системи, що мають

стандартизовані інтерфейси - вирішення проблеми відкритості ґрунтується на стандартизації інтерфейсів систем і протоколів взаємодії між їх компонентами. Крім того, ВІС має ряд необхідних для функціонування властивостей [3]:

- модульність - здатність апаратного забезпечення (АЗ) або ПЗ до модифікації шляхом додавання, видалення або заміни окремих компонентів системи без впливу на ІС в цілому;
- взаємозамінність - можливість заміни будь-якого модуля системи на аналогічний і можливість зворотної заміни;
- кросплатформеність - можливість роботи ІС різних апаратно-програмних платформах;
- інтероперабельність - здатність використовувати ПЗ, що експлуатується одночасно на різних операційних системах (ОС) в загальній мережі, з можливістю обміну інформацією між елементами цього ПЗ;

- масштабованість - можливість застосування одного і того ж АЗ і ПЗ для ІС різної розмірності.

Відкриті системи повинні мати стандартні інтерфейси для виконання вимог щодо можливості інтеграції з іншими системами. Використання ВІС має низку переваг:

- відкритість передбачає можливість простої інтеграції різнорідних систем;
- економія фінансових коштів - завдяки низькій вартості життєвого циклу;
- збільшений час безвідмовної роботи - завдяки можливості вибору найнадійніших модулів на ринку;
- мінімізований час вимушеного простою - завдяки великому вибору взаємозамінних модулів;
- мінімізація зусиль, необхідних для введення в дію ПЗ, - завдяки відсутності додаткового навчання;
- проста зміна конфігурації системи для роботи з новими технологічними процесами - завдяки властивостям модульності і розширюваності ВІС;
- застосування новітніх технологій і технічних рішень - завдяки широкому вибору рішень і спеціалізації виробників;
- збільшення часу життя системи - завдяки взаємозамінності і можливості нарощування функціональності.

Однак ВІС має також і недоліки [4]:

- при створенні ВІС відповідальність за працездатність системи в цілому лежить на системному інтеграторі, а не на виробникові системи;
- універсальні протоколи, інтерфейси, мережі і ПЗ повинні бути досить складними, і як наслідок, дорогими та ненадійними;
- іноді до ознак відкритості відносять відкритість вихідних кодів;
- спостерігається ефект зниження надійності ПЗ, частини якого пишуться різними виробниками;
- як і будь-яка стандартизація, відкритість накладає обмеження на діапазон можливих технічних рішень, знижуючи ймовірність появи нових.

За результатами аналізу існуючих недоліків ВІС можна зробити висновок, що в цілому проблема проектування ВІС не вирішена в достатній мірі. Особливо це відноситься до складності процесу розподілу відповідальності між виробником системи і системним інтегратором, що робить актуальним дослідження існуючих методів проектування ВІС і створення моделі ВІС, яка усуне даний недолік.

2. Аналіз існуючих стандартів і моделей інформаційних систем з відкритою архітектурою

В даний час розроблено понад двохсот міжнародних стандартів, які тою чи іншою мірою відображають концепції і методи створення та функціонування ВІС. При цьому основна увага акцентується або на стандартах, які підтримують переносимість програм, або на стандартах, які забезпечують комунікацію. У цих двох напрямках розробляються відповідні концепції і методи, які формалізуються і деталізуються комплексами стандартів. Для кожного напрямку характерний розвиток методів і стандартів, орієнтованих на підтримку і реалізацію конкретного напрямку, а також окремої групи методів і стандартів, загальних для обох напрямків.

Мета першого напрямку - створення методів відкритих систем (ОС), що забезпечують відносно простий і ефективний щодо трудомісткості і якості перенесення апробованих прикладних програмних засобів обробки інформації і баз даних (БД) на різні апаратні та операційні платформи [5].

Мета другого напрямку - забезпечення оперативного обміну даними між компонентами ІС, що базуються на гетерогенних і розподілених апаратних платформах (OSI). Для цього

потрібно було створити концепції та методи уніфікації оперативного транспортування даних між компонентами ІС в реальному часі. Проблема полягає в забезпеченні сумісності різних систем передачі даних і в ефективному використанні розподілених обчислювальних ресурсів для обробки інформації, а також в можливості попереднього вибору типів і ресурсів відповідно до потреб ІС для вирішення конкретних прикладних задач [6].

Основний економічний ефект в цьому випадку досягається за рахунок скорочення додаткових перетворень даних на стиках комунікаційних засобів і підвищення тим самим ступеня корисного використання обчислювальних ресурсів. Ці методи, так само, як і для першого напрямку, можна розділити на три частини: Порівняльна таблиця викладених вище напрямків представлена в табл. 1.

Таблиця 1

Назва	Мета	Групи методів	Стандарти	Переваги	Недоліки
OCS	Створення методів перенесення прикладних програмних засобів обробки інформації і БД на різні апаратні та операційні платформи	Локалізація та уніфікація інтерфейсів ПЗ	POSIX (концепція і функції інтерфейсів ОС, що переносяться)	Відносно просте створення системи	Необхідність створення або придбання хмарного сховища
		Уніфікація коду програм і опису даних	Стандарти забезпечують переносимість додатків	Можливість модифікації існуючої ІС	
		Уніфікація зовнішніх інтерфейсів додатків	Стандарти регламентують взаємодію з зовнішнім середовищем	Незалежність від програмної і апаратної платформ	
OSI	Забезпечення швидкого обміну даними між компонентами ІС	Створення загальної архітектури інтерфейсів ІС	Стандарти, що підтримують семирівневу модель	Поділ комунікації на простіші етапи	Неможливість використовувати при модифікації ІС
		Уніфікація основних системних функцій і зовнішніх інтерфейсів	Стандарти комунікації даних між компонентами ІС	Прискорення розробки нових програмних і апаратних продуктів	Шифрування даних тільки на високих рівнях моделі
		Уніфікація обміну інформацією в складних ІС	Стандарти регламентації взаємодії із зовнішнім середовищем	Забезпечення інтеграбельності	Складна реалізація

Припускаючи виконання організаційно-технічних та інших вимог до ВІС (таких як відкрита специфікація на інтерфейси, служби і формати даних), ввижається доцільним використовувати напрямок OCS, реалізований за допомогою об'єктно-орієнтованого підходу (ООП) для

формування базового набору модулів ІС з можливістю доповнення цих модулів необхідними замовнику функціями і структурами даних.

3. Постановка задачі дослідження

Основними оцінками ходу та результатів проектування ВІС, побудованих з використанням ООП, є мінімум складності міжоб'єктних інтерфейсів; максимум функцій взаємодії з існуючими платформами, реалізованих АЗ системи; мінімум складності компонент, що залежать від АЗ, або мінімум обсягу об'єктів, не задіяних при роботі в умовах конкретної платформи; максимум продуктивності компонент системи при вирішенні задач; максимум вірогідності обробки даних тощо.

На основі вихідних даних визначаються такі характеристики, як множина функціональних задач; множина процедур обробки даних; множина компонентів АЗ ВІС, що реалізують функціонування системи на різних апаратних платформах; варіанти можливої взаємодії процедур відкритої системи з АЗ; характеристики процедур обробки даних, прикладного ПЗ та АЗ і технічних засобів.

Як така, що сприяє досягненню мети дослідження, в роботі [7] пропонується модель вирішення задачі синтезу структури ВІС, що забезпечує складність компонент при функціонуванні системи на різних платформах. Ця модель має вигляд

$$\min_{\{x_{rv}, z_{vj}\}} \left\{ \sum_{r=1}^R \sum_{v=1}^V x_{rv} (1 - x_{r+1,v}) \left[C_v + \sum_{j=1}^R z_{vj} c_j \right] \right\}. \quad (1)$$

Але дана модель не враховує ряд обмежень, які можуть бути пред'явлені до системи, а також загальний обсяг компонент системи, що впливає на можливість включення в систему користувачів з будь-яким комплексом апаратних засобів. До того ж, дана модель не враховує складність міжоб'єктного інтерфейсу, що може позначитися на швидкості обміну інформацією між компонентами системи, як наслідок, впровадження подібної системи стає недоцільним. Це призвело до необхідності доопрацювання і вдосконалення моделі рішення задачі синтезу структури ВІС. Таким чином, були виділені такі задачі даного дослідження:

- удосконалення моделі вирішення задачі синтезу структури ВІС;
- аналіз і вибір алгоритмів вирішення задачі синтезу структури ВІС;
- апробація отриманих результатів.

4. Удосконалення моделі вирішення задачі синтезу структури відкритої інформаційної системи

Приймемо як оцінку оптимальності ВІС, експлуатація якої здійснюється на різних платформах, відношення загального обсягу компонент, використовуваних в ході функціонування системи на конкретній апаратній платформі, до загального обсягу компонентів системи. Тоді задачу проектування ВІС можна сформулювати таким чином:

$$\min_{\left\{ \begin{matrix} x_{rv} \\ x_{rp} \\ y_{if} \end{matrix} \right\}} \left(\sum_{v=1}^V \sum_{r=1}^R \sum_{p=1}^R \left\{ x_{rv} x_{rp} \left(1 - \sum_{r'=1}^R x_{r'v} x_{r,p+1} \right)^* \right. \right. \\ \left. \left. * \left[H_v + \sum_{f=1}^F \left(q_j K_f z_{vl}^{rp} + \gamma_{vl}^{rp} \sum_{l=1}^L k_l a_{vl}^{rp} y_{lf} \right) \right] \right\} \right)^* \\ * \left(\sum_{v=1}^V \sum_{r=1}^R \sum_{p=1}^R \left\{ x_{rv} x_{rp} \left(1 - \sum_{r'=1}^R x_{r'v} x_{r,p+1} \right)^* \right. \right. \\ \left. \left. * \left[H_v + \sum_{f=1}^F \left(C_v + \sum_{j=1}^F c_j (z_{vl}^{rp} + \varepsilon_{vl}^{rp}) \right) \right] \right\} \right)^{-1} \quad (2)$$

при обмеженнях:

- на загальне число об'єктів в системі $V \leq \bar{V}$, де \bar{V} - максимально допустима кількість об'єктів в системі;

- на включення окремих процедур до складу одного об'єкта $x_{rv} + x_{\bar{r}v} \leq 1$, $x_{rv} x_{\bar{r}v} x_{r\bar{v}} \dots = 1$ для заданих процедур $a_{rv} a_{\bar{r}v} a_{r\bar{v}} \dots$; $V = 1, 2, \dots, \bar{V}$;

- на загальне число процедур або груп процедур в складі кожного об'єкта $\sum_{r=1}^R x_{rv} \leq \overline{M}_v$,

$\sum_{r=1}^R (x_{rv} x_{\bar{r}v} x_{r\bar{v}} \dots) \leq \tilde{M}_v$, де M_v - максимально допустима кількість процедур в v-му об'єкті;

\tilde{M}_v - максимально допустима кількість груп процедур в v-му об'єкті;

- на одноразовість включення процедур обробки даних в об'єкти $\sum_{v=1}^V x_{rv} = 1, r = 1, 2, \dots, R$;

- на передачу управління з об'єкта до завершення обробки даних усіма процедурами

об'єкта $\sum_{r=1}^R \sum_{p=1}^R x_{rp} x_{rp} \left(1 - \sum_{\bar{r}=1}^R x_{\bar{r}v} \right) = 1$ для заданого v-го об'єкта;

- на обсяг кожного об'єкта $\sum_{r=1}^R h_r x_{rv} \leq \overline{H}_v, v = 1, 2, \dots, V$ де \overline{H}_v - максимально допустимий обсяг v-го об'єкта;

- на число системних функцій, що викликаються процедурами кожного об'єкта

$\sum_{l=1}^L z_{vl} \leq L_v, v = 1, 2, \dots, V$, де $z_{vl} = 1$, якщо $\sum_{r=1}^R w_{rl} w_{rv} = 1$, і $z_{vl} = 0$ в іншому випадку; L_v - максимально припустима кількість системних функцій, що викликаються процедурами v-го об'єкта;

- на складність інтерфейсу між об'єктами системи $\sum_{l=1}^L \sum_{v=1}^{V-1} \sum_{v'=v+1}^V z_{vl} z_{v'l} \leq S$, де S - максимально припустимий інформаційний інтерфейс між об'єктами системи;

- на складність інформаційного інтерфейсу між окремо взятими об'єктами системи

$\sum_{l=1}^L z_{vl} z_{v'l} \leq S_{v'v}$, де для заданих об'єктів v і v' $S_{v'v}$ - максимальне число загальних входів-виходів об'єктів;

- на дублювання апаратно-залежних компонентів системи $\sum_{f=1}^F y_{gf} = g, g = 1, 2, \dots, G$, де G - припустима ступінь дублювання апаратно-залежних компонентів системи, тобто реалізації в компонентах АЗ функціонування системи на одній платформі;

- на число компонентів АЗ ВІС, загальних для компонентів прикладного ПЗ

$\sum_{f=1}^F \sum_{v=1}^{V-1} \sum_{v'=v+1}^V z_{vf} z_{v'f} \leq C$, де C - максимально допустима кількість компонентів АЗ ВІС, що є загальними для виділених об'єктів прикладного ПЗ;

- на число компонентів АЗ ВІС, загальних для окремих об'єктів прикладного ПЗ $\sum_{f=1}^F z_{vf} z_{v'f} \leq C_{v'v}$,

де для заданих об'єктів v і v' $C_{v'v}$ - максимальне число загальних входів-виходів об'єктів;

- на одноразовість виконання кожної процедури в послідовності обробки даних

$\sum_{p=1}^P x_{rp} w_{rp} = 1, r = 1, 2, \dots, R$;

- на число одночасно виконуваних процедур $\sum_{p=1}^P x_{rp} w_{rp} = a, r = 1, 2, \dots, R$; де a - допустиме число одночасно виконуваних процедур в послідовності виконання.

5. Вибір методів вирішення задачі синтезу структури відкритої інформаційної системи

Задача синтезу системи об'єктів прикладного ПЗ, що мінімізує інформаційний інтерфейс, є нелінійною задачею цілочисельного програмування. Для вирішення задач даного типу можливе використання таких методів:

- метод можливих напрямків;
- метод Келлі, або метод січних площин;
- алгоритм Гомори;
- метод гілок і меж;
- метод штрафів;
- метод спрямованого конструювання рішення з обмеженим перебором [8].

На основі аналізу даних методів як найбільш доцільні для вирішення задач оптимізації були обрані метод гілок і меж і метод спрямованого конструювання рішення з обмеженим перебором. Далі на основі вибраних методів був розроблений наступний алгоритм вирішення задачі.

Крок 1. У процесі розгалуження для деякої вершини фіксується множина змінних $\{\bar{x}_{rv}\}$.

Крок 2. У множині $\{\bar{x}_{rv}\}$ визначається змінна $x_{r^*v^*}$ з максимальним індексом по r , тобто $r^* = \max\{r\}$.

Крок 3. Для об'єкта з індексом v^* обчислюється його вільний об'єм $d_v = \bar{M}_{v^*} - \sum_{d_v \in M} x_{rv^*}$, де \bar{M} - множина процедур в складі об'єкта з індексом v^* .

Крок 4. Якщо $d_v \neq 0$, фіксується частина множини змінних доповнення, що забезпечують нижню межу рішень. У цю множину входять змінні з індексом $r = r^* + 1, r^* + 2 \dots$ і $v = v^*$.

Крок 5. Обчислюються вільні обсяги $d_v, v = \overline{1, V^*}$ для всіх об'єктів, в які розподілено хоча б одну процедуру.

Крок 6. Обчислюється загальна кількість нерозподілених процедур $C = R - \sum_{r=1}^R \sum_{v=1}^{V^*} \bar{x}_{rv}$, де V^* - число вже утворених об'єктів прикладного ПЗ; якщо $C = 0$ - перехід до кроку 9.

Крок 7. Якщо $C = \max\{d_v\}$ і $V^* \leq V$, фіксується множина змінних $\{\bar{x}_{rv}\}$, де V - максимально припустима кількість утворюваних об'єктів прикладного ПЗ; \bar{r} - індекс процедури, розподіленої останньою в процесі роботи алгоритму.

Крок 8. Якщо $C = \max\{d_v\}$ і $V^* = V$ фіксується множина змінних $\{\bar{x}_{rv}\}$, починаючи з об'єкта максимального вільного об'єму в порядку зменшення їх значень. Перехід до кроку 5.

Крок 9. Обчислюється значення величини $\bar{n}_i^k(\bar{x}_{rv})$ відповідно до виразу

$$\bar{n}_i^k = \sum_{r=1}^R \sum_{v=1}^V x_{rv}(1 - x_{r+1,v}).$$

Таким чином, розроблений алгоритм включає такі основні операції:

- визначення $\{y_{\theta_v}\}$, відповідних до мінімальної оцінки;
- побудова альтернативних мереж G_v у верхівці дерева розгалуження з мінімальною оцінкою;
- впорядкування мереж G_v відповідно до ваги p в порядку убавання ваг;
- визначення для кожної мережі оптимального шляху, що складається з набору процедур.

6. Апробація результатів використання методу проектування інформаційної системи з відкритою архітектурою

Апробація результатів проводилася на прикладі інформаційно-аналітичної системи "Реєстр Ліги українських клубів (ЛУК)" (ІАС "Реєстр ЛУК"). Ця система розроблялася на замовлення Всеукраїнської громадської організації "ЛУК інтелектуальних ігор". Метою створення ІАС "Реєстр ЛУК" було підвищення ефективності роботи зазначеної організації шляхом

автоматизації функціональних задач ІАС, що забезпечує зниження часових витрат організаторів та учасників ігор на пошук потрібної інформації; автоматизація документообігу організації; формування статистичних звітів для аналізу та планування ігор; а також організація веб-доступу до актуальної інформації про діяльність ЛУК та її учасників для підвищення оперативності взаємодії учасників та організаторів ігор [9,10].

Розробка і впровадження ІАС "Реєстр ЛУК" як ВІС реалізовано з використанням методів побудови структури та об'єктів прикладного ПЗ, викладених вище. Як критерії синтезу структури прикладного ПЗ були використані: мінімум складності міжоб'єктного інтерфейсу для кожної підсистеми і мінімум обсягу компонентів системи в цілому. Як критерії синтезу об'єктів прикладного ПЗ використовувалися: мінімум загального числа звернень об'єктів один до одного в процесі функціонування системи, мінімум технологічної складності реалізації взаємодії об'єктів з компонентами АЗ і з об'єктами, що містять процедури обробки даних, які залежать від конкретної реалізації.

Для вирішення задач синтезу були виділені об'єкти, реалізовані кожною підсистемою, що входить до ІАС "Реєстр ЛУК", їх вхідні і вихідні дані, зроблено оцінку складності та обсягу реалізації процедур, а також обсягу переданих даних. Компоненти системи і їх скорочені найменування наведено в табл. 2; об'єкти, що входять до складу компонентів системи, представлені в табл. 3.

Таблиця 2

Умовне найменування	Компонент
ОГ	Облік гравців
ОК	Облік команд
ОСК	Облік складів команд
ОЧ	Облік членства ЛУК
ОПЗ	Облік поданих заявок на захід
ОСВ	Облік сплати вступних і членських внесків
ОСІВ	Облік сплати ігрових внесків
ОТ	Облік тарифів вступних і членських внесків
ОП	Облік посад ЛУК
ОЗП	Облік посад, займаних членами ЛУК

Як видно з табл. 3, при роботі багатьох прикладних підсистем використовуються ті ж самі об'єкти інформаційного забезпечення ІАС "Реєстр ЛУК". Ця обставина викликає необхідність багаторазової реалізації прикладного ПЗ на етапах вибору і завантаження даних, пошуку, збереження результату тощо. При цьому кожна подібна реалізація повинна бути модифікована для вирішення конкретної функціональної задачі. Будь-які зміни структури прикладного ПЗ, неминучі в процесі еволюції системи, повинні бути одночасно доступними для всіх спеціалізованих підсистем після реалізації цих змін в об'єктах одного або декількох рівнів ієрархічної структури.

Як було викладено вище, задача синтезу структури ВІС може бути вирішена з врахуванням ряду обмежень. Конкретні значення правих частин цих обмежень для вирішення задачі побудови ІАС "Реєстр ЛУК", наведено в табл. 4. Обсяг пам'яті, яка потрібна для реалізації кожного з об'єктів системи, представлено в табл. 5.

Для вирішення задачі використовувався алгоритм, заснований на спільному застосуванні методів гілок і меж і спрямованого конструювання рішення. В результаті розрахунку за запропонованим алгоритмом було отримано оптимальне рішення задачі, яке пропонує розділити систему на п'ять окремих компонентів (функціональних задач). Опис входження об'єктів системи до відповідних компонентів наведено в табл. 6.

Отримане рішення забезпечує необхідний ступінь розширення системи, можливостей для її адаптації до умов, що змінюються в процесі еволюції завдання обробки даних, уніфікації інформаційних інтерфейсів. Завдяки комплексному об'єктно-орієнтованому проектуванню з використанням методик проектування прикладного ПЗ ВІС була отримана

Таблиця 3

Об'єкт	ОГ	ОК	ОСК	ОЧ	ОПЗ	ОСВ	ОСІВ	ОТ	ОП	ОЗП
Гра					+					
Гравець	+	+	+	+	+	+	+			+
Ігровий регіон	+	+	+	+	+	+	+			+
Ліга					+		+			
Рівень складності ліги					+		+			
Склад ліги					+		+			
Сезон					+		+			
Чемпіонат					+		+			
Легионер			+		+					
Капітан	+	+	+		+					
Фізична особа	+	+	+	+	+	+				+
Юридична особа	+	+	+	+	+	+				
Команда		+	+		+		+			
Склад команди			+		+					
Член ЛУК				+	+	+				+
Вікова категорія				+	+	+		+		
Внесок				+	+	+		+		
Тариф								+		
Контракт										+
Вакансія										+
Посада									+	+
Захід					+		+			
Заявка на захід					+		+			

Таблиця 4

Обмеження	Значення
На загальне число об'єктів в системі	5
На включення окремих процедур до складу одного об'єкта	-
На загальне число процедур або груп процедур в складі кожного об'єкта	7
На одноразовість включення процедур обробки даних в об'єкти	-
На обсяг кожного об'єкта	2048КБ
На число системних функцій, що викликаються процедурами кожного об'єкта	-
На складність інтерфейсу між об'єктами системи	32
На дублювання об'єктів системи	1
На число компонентів АЗ ВІС, загальних для компонентів ПЗ	-
На одноразовість виконання процедури в послідовності обробки даних	-
На число одночасно виконуваних процедур	-

структура об'єктів, що виконують функціональні задачі із заданою ефективністю, досягнута інтероперабельність системи, мобільність системи і персоналу.

7. Висновки

В ході дослідження було вдосконалено модель задачі синтезу структури ВІС за запропонованими критеріями ефективності. Поставлена задача є задачею нелінійного цілочисельного програмування комбіаторного типу, для вирішення якої запропоновані алгоритми, засновані на використанні методу гілок і меж.

Таблиця 5

Позначення	Об'єкт	Обсяг, Кб
a ₁	Гра	512
a ₂	Гравець	512
a ₃	Ігровий регіон	256
a ₄	Ліга	256
a ₅	Рівень складності ліги	128
a ₆	Склад ліги	1024
a ₇	Сезон	256
a ₈	Чемпіонат	256
a ₉	Легіонер	128
a ₁₀	Капітан	128
a ₁₁	Фізична особа	128
a ₁₂	Юридична особа	128
a ₁₃	Команда	256
a ₁₄	Склад команди	1024
a ₁₅	Член ЛУК	128
a ₁₆	Вікова категорія	256
a ₁₇	Внесок	128
a ₁₈	Тариф	256
a ₁₉	Контракт	128
a ₂₀	Вакансія	128
a ₂₁	Посада	128
a ₂₂	Захід	256
a ₂₃	Заявка на захід	512

Таблиця 6

Компонент	1	2	3	4	5
Загальний об'єм	1024	1536	1920	1 280	1 280
Об'єкт	a ₂ , a ₃ , a ₁₁ , a ₁₂	a ₉ , a ₁₀ , a ₁₃ , a ₁₄	a ₁ , a ₄ , a ₅ , a ₆	a ₇ , a ₈ , a ₂₂ , a ₂₃	a ₁₅ , a ₁₆ , a ₁₇ , a ₁₈ , a ₁₉ , a ₂₀ , a ₂₁

articles/art_19.shtml. 4. *Матяш С.А.* Корпоративні інформаційні системи // Directmedia. 2016. №1 (11). С. 161-164. 5. *Vogel B., Gkouskos D.* An open architecture approach: Towards common design principles for an IOT architecture // ACM International Conference Proceeding Series. 2017. Part F130530. P. 85-88. doi: 10.1145 / 3129790.3129793. 6. *Bagby P., Hubert A.J., White R., Cafarelli S., Shaver J.* Open architecture modeling, verification, and validation // AIAA Modeling and Simulation Technologies Conference. 2018. 7. *Яблонський А.С.* Формальна методологія проектування відкритих систем // X міжнародна конференція "Проблеми управління безпекою складних систем". М. 2007. Т.2. С.140-142. 8. *Function (X).* Рішення задач цілочисельного програмування: методи і приклади. Режим доступу: https://function-x.ru/zadacha_celochislenogo_programirovaniya.html. 9. *ЛУК.* Про Лігу українських клубів інтелектуальних ігор. Режим доступу: <http://luk.org.ua/about>. 10. *Євланов М.В.* Моделі, методи і інформаційна технологія розробки архітектури складних інформаційних систем на основі функціональних вимог: дис. ... д-ра техн. наук. Харків: ХНУРЕ, 2017. 480 с.

Запропоновано постановку і методи вирішення задачі створення прикладного ПЗ ВІС на етапі технічного проектування. В результаті вирішення розглянутої задачі визначаються оптимальний склад і число об'єктів прикладного ПЗ, а також склад ВІС.

Апробація результатів дослідження проводилася під час проектування структури компонентів ІАС "Реєстр ЛУК". Результати апробації дозволяють максимально уніфікувати АЗ і прикладне ПЗ ІВС, досягти необхідного рівня мобільності функціональних підсистем ІАС, їх інтегровуваності, спростити обслуговування та модифікацію ІАС.

Список літератури: 1. *Інформаційні технології в економіці.* Основні проблеми створення і використання інформаційних систем та інформаційних ресурсів організації. Режим доступу: http://magmenit.narod.ru/essays/essay_1.html. Тема з екрану. 2. *Кунгурцева В.С., Титов А.Б.* Тенденції та проблеми інноваційного розвитку інформаційно-комунікаційних систем в умовах цифрової економіки // Науково-технічні відомості Санкт-Петербурзького державного політехнічного університету. Економічні науки. 2018. №1 (11), С. 54-63. doi: 10.18721 / JE.11105. 3. *Відкриті системи, процеси стандартизації та профілі стандартів.* Режим доступу <http://citforum.ck.ua/database/>

Надійшла до редколегії 21.06.2018

Карабицька Олена Дмитрівна, студентка кафедри ІУС. Наукові інтереси: проблеми інтелектуалізації проектування і експлуатації інформаційних управляючих систем. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 70-21-451.