

УДК 004

М.Красівський, магістр гр. КН-22М-2

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕАЛІЗАЦІЇ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ОРГАНІЗАЦІЇ ПАКЕТІВ ПРИКЛАДНИХ ПРОГРАМ

У статті розроблено програмне забезпечення, яке призначено для системи реалізації інструментальних засобів організації пакетів прикладних програм. Метою розробки є дослідження та програмна реалізація системи реалізації інструментальних засобів організації пакетів прикладних програм. Об'єктом дослідження є процес реалізації інструментальних засобів організації пакетів прикладних програм. Предметом дослідження є методи реалізації інструментальних засобів організації пакетів прикладних програм. Методи дослідження базуються на методах інженерії програмного забезпечення, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи реалізації інструментальних засобів організації пакетів прикладних програм. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. У більшості організацій команда пакування програм також є частиною розширеної команди SCCM. Я бачив багато пакувальників програм, які хотіли вивчити SCCM та Intune.

Пакування додатків – це процес створення пакетів програмного забезпечення для легкого встановлення на пристроях користувача. Application Packaging Tool – це утиліта програмного забезпечення, яка допомагає підготувати та розповсюдити програми для розгортання.

Упакування додатків може принести значну користь компаніям скоротивши витрати, зменшивши збої та підвищивши ефективність у різних відділах.

Пакування програми спрощує розгортання програмного забезпечення, знижує витрати, підвищує безпеку та стабільність, а також полегшує керування на різних платформах.

Наведемо основні переваги упаковки додатків:

1. Стандартизація та постійність: упаковка додатків створює узгоджене IT-середовище.
2. Зменшені витрати на підтримку: упаковка мінімізує потенційні збої в програмі, що призводить до зменшення кількості дзвінків у службу підтримки.
3. Розповсюдження по всьому світу: упакуйте один раз і розгорніть всюди.
4. Мінімальне заважання: безшумне встановлення забезпечує безперешкодну роботу систем.
5. Просте керування: спрощене керування IT-додатками для зручності користувача.
6. Розширене звітування: вичерпна інформація про програми для швидкого усунення несправностей.
7. Покращена безпека: безпечніше IT-середовище з меншими правами адміністратора.
8. Заборонене встановлення програмного забезпечення: більше жодних незаконних встановлення.
9. Індивідуалізація: стильна упаковка, що відповідає точним потребам бізнесу.

10. Найкраще оформлення упаковки: узгодження з найкращими світовими практиками.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-20] було виявлено певні прогалини у забезпеченні системи реалізації інструментальних засобів організації пакетів прикладних програм.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи реалізації інструментальних засобів організації пакетів прикладних програм.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем реалізації інструментальних засобів організації пакетів прикладних програм.

- Дослідження системи реалізації інструментальних засобів організації пакетів прикладних програм.

- Програмна реалізація системи реалізації інструментальних засобів організації пакетів прикладних програм.

Об'єктом дослідження є процес реалізації інструментальних засобів організації пакетів прикладних програм.

Предметом дослідження є методи реалізації інструментальних засобів організації пакетів прикладних програм.

Методи дослідження базуються на методах інженерії програмного забезпечення, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Розглянемо основні поняття, пов'язані з організацією розподілених пакетів прикладних програм (РППП), формалізований опис предметної області РППП, розроблені мовні засоби подання опису предметної області РППП у форматі XML і сформульовані вимоги до інструментального комплексу, призначеному для створення й застосування РППП у різноманітних розподілених обчислювальних середовищах (РОС).

РППП орієнтуються на клас завдань, що характеризуються наступними властивостями:

- рішення завдання вимагає проведення розрахунків на ЕОМ з використанням більших обсягів обчислювальних ресурсів (процесорного часу, оперативної пам'яті, дискового простору й ін.);

- можлива декомпозиція загального складного завдання на більше прості (з обчислювальної точки зору) підзадачі;

- процес рішення загального завдання має на увазі розподілене рішення набору її взаємозалежних підзадач;

- не передбачається інтенсивної взаємодії між паралельними процесами рішення підзадач;

- завдання допускає декомпозицію даних на блоки й незалежну паралельну обробку цих блоків;

- рішення завдання виконується, як правило, по одній слабко змінюваній синхронній схемі.

До такого класу відносяться завдання, що вимагають проведення різноманітних розрахунків, а також багато переборних завдань.

Нижче наведені характерні відмінності середовища функціонування, мовних засобів, функціонального наповнення й системної частини РППП від відповідних складових традиційних пакетів:

- Середовище функціонування. Середовищем функціонування РППП є РОС із різноманітними обчислювальними вузлами, організованими на основі різних програмно-апаратних платформ і керованими різними ОС. Найчастіше різноманітні РОС мають низький ступінь відказостійкості.

– Мовні засоби. Як вхідна мова пакета використовується розширювана мова розмітки XML. Потрібні засоби опису великоблочного розподіленого алгоритму рішення завдання у РОС.

– Функціональне наповнення РППП. Як модулі використовуються програми, що виконуються в пакетному режимі, які можуть бути написані на різних мовах програмування (наприклад, C#, Delphi XE5 і ін.), функціонувати під управлінням різних ОС (наприклад, MS Windows, Linux, Mac OS X і ін.) і бути платформи-залежні. Модулі розміщуються в різних вузлах РОС, в одному вузлі РОС може бути встановлено кілька модулів. Припустимо включення до складу функціонального наповнення успадкованого ПЗ, а також нетиражованих програмних комплексів, жорстко прив'язаних до вузлів РОС. Обмін даними між модулями здійснюється через файли.

– Системна частина. Потрібна організація обчислень на основі віддаленого запуску модулів і розподіленого обміну даними. Необхідний моніторинг вузлів РОС.

Більшість інструментальних систем (наприклад, OLYMPUS, ПРИЗ, САФРА, СУПЕРЕЧКИ й ін.) розроблялися, як правило, для створення традиційних ППП. Застосування таких систем для побудови РППП є скрутним. Відомі системи організації розподілених обчислень (наприклад, кластерна система Condor, програмний комплекс BOINC, інструментарій X-COM і др.) дозволяють здійснити у РОС рішення не зв'язаних між собою завдань, що допускають розпаралелювання по даним, але не мають всі необхідні можливості для реалізації перерахованих вище особливостей розподіленого пакета взаємозалежних прикладних програм. У цьому зв'язку виникає необхідність розробки інструментарію для організації спеціалізованого виду прикладних програмних комплексів – розподілених пакетів прикладних програм (РППП).

Властивості класу завдань, характерного для РППП (зокрема, рішення завдання по одній слабко змінюваній схемі), спричиняються вибір процедурного способу постановок завдань пакету. Ці ж властивості в сукупності з особливостями функціонального наповнення РППП (використання як модулі програм, що виконуються в пакетному режимі й розміщених у різних вузлах РОС) приводять до необхідності реалізації процесу рішення завдання по заданій синхронній схемі в режимі інтерпретації.

Формалізований опис предметної області РППП відноситься до класу обчислювальних моделей і являє собою сукупність значимих параметрів предметної області, а також модулів РППП, що реалізують обчислювальні операції із цими параметрами.

У найпростішому випадку опис предметної області РППП можна визначити у вигляді структури:

$$S = \langle Z, T, M, Y \rangle,$$

де Z – множина параметрів, T – множина припустимих типів параметрів, M – множина модулів, Y – множина вузлів РОС, у яких розміщений той або інший модуль із M . Зв'язку між елементами множин Z , T , M і Y задані відносинами $ZT \subset Z \times T$, $IN \subset Z \times M$, $OUT \subset Z \times M$ і $MY \subset M \times Y$ (у загальному випадку типу «багато-к-багатьох»). Для кожного об'єкта структури S (параметра, типу параметра, модуля й вузла) визначений набір його атрибутів. Множина T включає наступні типи параметрів: тип *file*, використовуваний для опису параметрів невизначеної структури (блоків довільного тексту великого розміру); тип *filelist*, призначений для підтримки розпаралелювання по даним (паралельний список параметрів типу *file*); тип *fileconst*, уведений з метою скорочення обсягів передачі даних у РОС (значення параметра типу *fileconst* один раз передається вузлу РОС і потім може неодноразово використовуватися при запуску модулів, розміщених у даному вузлі). Змістовно модуль $m_i \in M$ реалізує можливість обчислення множини параметрів $OUT \subset Z$ по множини параметрів $IN^i \subset Z$, а множини IN^i і OUT^i називаються відповідно множинами вхідних і вихідних параметрів для модуля m_i . Заелементна обробка параметра z_j типу *filelist* модулем m_i виконується в такий спосіб: k -й елемент параметра z_{jk} обробляється k -м екземпляром модуля m_{ik} . Опис предметної області РППП будемо вважати цілісним, якщо виконуються наступні умови:

1. Умова наявності опису предметної області:

$$(Z \neq \emptyset) \wedge (M \neq \emptyset).$$

2. Умова наявності вхідних і вихідних параметрів модулів:

$$\forall m_i \in M (IN^i \neq \emptyset) \wedge (OUT^i \neq \emptyset).$$

3. Умова цілісності відносин IN і OUT :

$$\forall m_i \in M IN^i \cup OUT^i \subset Z.$$

4. Умова відсутності в модулях транзитних параметрів:

$$\forall m_i \in M IN^i \cap OUT^i = \emptyset.$$

5. Умова ненадмірності множини параметрів:

$$\forall z_j \in Z \exists m_i \in M : z_j \in IN^i \cup OUT^i.$$

Постановка завдання для структури S задається в процедурному виді й у загальному випадку формулюється в такий спосіб: «знаючи S виконати Q », де $Q \subset M$ являє собою частково впорядковану послідовність модулів з M , які необхідно виконати для рішення завдання. При встановленні часткового порядку множина Q розбивається на до непустих підмножин. Упорядкування підмножин здійснюється залежно від того, модулі якої підмножини повинні бути виконані раніше. У рамках кожної k -ї підмножини вхідні в нього модулі можуть виконуватися незалежно друг від друга в будь-якій послідовності або паралельно.

Схемою рішення завдання (CPЗ) будемо називати модель великоблочної програми, що відбиває інформаційно-логічну структуру обчислень у термінах предметної області. CPЗ будується в паралельно-ярусній формі з елементів наступних множин:

- множини параметрів Z ; множини модулів M ;
- множини подій E , що виникають у процесі виконання CPЗ;
- множини функцій F , призначених для управління процесом виконання CPЗ;
- множини спеціальних операторів:

$O = \{START, STOP, READ \langle \text{список параметрів} \rangle,$

$WRITE \langle \text{список параметрів} \rangle,$

$CALL \langle \text{ім'я модуля} \rangle,$

$FORK,$

$JOIN,$

$TERMINATE \langle \text{список модулів} \rangle\}$.

Інформаційно-логічні зв'язки між параметрами й модулями CPЗ представляються у вигляді структури $H = \langle IN^h, \cup, OUT^h \rangle$, де IN^h і OUT^h – множини вхідних і вихідних параметрів CPЗ, відповідно. Структура H зображується двочастковим орієнтованим інформаційним графом G_h , що включає дві непересічні множини вершин: вершин-параметрів і вершин-модулів. Вершини-параметри, як і вершини-модулі, є попарно несуміжними. Частково впорядковану послідовність модулів Q зручно задавати у вигляді булевої матриці розмірності $k \times n$, де k – число ярусів CPЗ, а n – число модулів в M . Елемент матриці $q = 1$ означає, що модуль m розміщений на l -ому ярусі CPЗ.

Побудова CPЗ здійснюється у два етапи:

- розроблювач РППП формує яруси CPЗ, розміщає на цих ярусах модулі з M і визначає події з E , при виникненні яких необхідно виконати ті або інші керуючі функції з F ;
- транслятор постановки завдання на основі інформаційно-логічних зв'язків між параметрами й модулями структури S визначає множини вхідних і вихідних параметрів CPЗ і доповнює постановку завдання спеціальними операторами з O . CPЗ будемо називати припустимою, якщо вона задовольняє наступним умовам:

1. Умова процедурної постановки завдання:

$$(Q \neq \emptyset) \wedge (IN^h \cup OUT^h = \emptyset).$$

2. Умова допустимості включення модуль у CPЗ:

$$\forall m_i \in Q : q_{pi} = 1 \quad IN^i \setminus (IN^h \cup (\bigcup_{j=1}^{p-1} \bigcup_{\substack{l=1 \\ \forall q_{jl}=1}}^n OUT^l)) = \emptyset.$$

3. Умова інформаційної незалежності модулів, розташованих на тому самому ярусі СРЗ:

$$\forall m_i \in Q : q_{pi} = 1 \quad IN^i \cap (\bigcup_{\substack{j=1 \\ (\forall q_{pj}=1) \wedge (j \neq i)}}^n OUT^j) = \emptyset.$$

Керуючим графом СРЗ будемо називати впорядкований орієнтований граф O_u , вершинами якого є оператори виклику модулів з M (або їхніх екземплярів), а також спеціальні оператори.

Кожна дуга v_j графа O_u характеризується парою величин $(/; c)$, де $/ \in \{0,1\}$ – це логічний перемикач, що визначає можливість ($/=1$) або неможливість ($/=0$) переходу по дузі v_j , а c – обмежник числа переходів по дузі v_j . Запис $(; c)$ для дуги v_j означає безумовний перехід по даній дузі.

Виконання СРЗ у режимі інтерпретації являє собою процес послідовно-паралельного виконання її операторів відповідно до порядку їхнього входження в граф O_u . Передача даних (значень параметрів) між модулями виконується відповідно до графа O . Виконання СРЗ будемо вважати *коректним*, якщо воно завершується оператором *STOP* за кінцеве число кроків.

Опис предметної області РППП і постановок завдань представляються у форматі, розробленому на основі розширюваного метамови XML. Функціональні й системні вимоги до інструментального комплексу DISCOMP, призначеному для створення й застосування РППП у різномірних РОС, сформульовані на основі аналізу відомих методів і засобів організації розподілених обчислень.

Основні тенденції в розвитку ППП

Як висновок відзначимо перспективні напрямки подальшого розвитку прикладного ПЗ. На сьогоднішній день як основних факторів, що впливають на функціональність ППП і складність їхньої розробки ПЗ, можна відзначити наступні:

- ріст продуктивності персональних комп'ютерів;
- розширення класів розв'язуваних завдань;
- збільшення загального числа користувачів;
- значна кількість раніше створеного (успадкованого) ПЗ;
- розвиток Інтернет і корпоративні мережі.

Розробка додатків з урахуванням цих факторів привела до появи прикладних пакетів і інтегрованих середовищ, які по своїх характеристиках виходять за рамки ППП четвертого покоління. Серед відмітних рис ПЗ нового покоління наступні:

- інтеграція компонентів прикладного пакета не тільки з додатками пакета, але й з оточенням;
- широке використання галузевих стандартів;
- використання інфраструктури Інтернет;
- платформонезалежність.

Особливу значимість на подальшій сценарій розвитку ППП має вплив технологій Інтернет і, зокрема, Web. Можливості, що представляються глобальною мережею дозволяють обмінюватися будь-якою інформацією, яку можна представити в цифровому виді. Це вже зараз із успіхом використовується в провідних пакетах прикладних програм, у першу чергу для забезпечення спільної роботи користувачів. Практична реалізація загального доступу можлива, наприклад, з використанням проміжного ПЗ (middleware). Так, при використанні технології Active, у документ MS Word або таблицю MS Excel можна помістити будь-який документ, що підтримує Active. Впровадженням може бути документ,

розміщений в Інтернет, більше того, є потенційна можливість відредагувати його й зберегти зміни в Мережі.

Все більшою популярністю користується концепція «тонких клієнтів». Під «тонким клієнтом» мається на увазі Інтернет-браузер. Сучасні браузери дозволяють відображати не тільки гіпертекстові документи, але й зображення в растрових і векторних форматах, відео- і аудіодані. Крім цього, браузери представляють засоби інтерактивної взаємодії з веб-серверами у вигляді різних веб-форм (від форм авторизації або пошуку до форм завантаження файлів) і підтримують виконання програм-скриптів на своїй стороні. Це дозволяє створювати програми, що завантажуються з веб-сервера, але виконувані в браузері. Прикладом такого рішення є сервіси Google Docs (Google Документи). Користувачам цього сервісу представляється можливість створювати й редагувати текстові документи, електронні таблиці й презентації прямо у вікні браузера, зберігати їх в Інтернет і надавати в спільне використання.

Через повсюдне проникнення Інтернету, можна говорити про те, що прикладне програмне забезпечення буде переходити в розряд сервісу, тобто користувачі будуть працювати з необхідним програмним забезпеченням через Мережу, одержуючи на свої комп'ютери готові результати. Отже, необхідність у більших локальних потужностях частково відпаде, що буде сприяти росту попиту на недорогі комп'ютери з низьким енергоспоживанням.

В основі технологій, що забезпечують подібні можливості, ряд спільних наробітків провідних виробників ПЗ і організацій по стандартизації. До них відносяться сервісно-орієнтована архітектура корпоративних додатків (веб-сервіси) і стандартизовані формати документів.

Веб-сервіси

Веб-сервіс – програмна система, доступна через локальну або глобальну мережу по заданій адресі, чії загальнодоступні інтерфейси визначені мовою XML. Ця програмна система доступна іншим програмними системами, які можуть взаємодіяти з нею за допомогою XML-повідомлень, переданих за допомогою інтернет-протоколів. Веб-служба є одиницею модульності при використанні сервісно-орієнтованої архітектури додатка. Іншими словами, веб-сервіс – це іменованний компонент розподіленої прикладної системи, доступний по гіпертекстових протоколах.

Основними достоїнствами веб-сервісів є:

- інтероперабельність;
- відкритість архітектури;
- взаємодія програмних систем через засоби захисту інформації (проксі-сервери, міжмережні екрани).

Основним недоліком є менша продуктивність додатків і більший обсяг мережного трафіку в порівнянні з іншими технологіями розподілених обчислень (RMI, CORBA, DCOM/Active). Ще одним недоліком є підвищена вимогливість до апаратних ресурсам на стороні сервера додатків (постачальника веб-сервісів).

Уніфікація форматів

Перспективним напрямком у розвитку ППП є використання уніфікованих форматів документів на основі відкритих стандартів. Відкритий стандарт – загальнодоступна специфікація, звичайно розроблювальна некомерційною організацією по стандартизації, вільна від ліцензійних обмежень при використанні. Відкриті формати є підмножиною відкритих стандартів і визначають специфікації зберігання й подання цифрових даних.

Використання відкритих форматів у ППП дозволяє гарантувати можливість доступу до даних з будь-якого сумісного додатка без оглядки на ліцензійні права й технічні специфікації. Актуальність концепції відкритих форматів підтверджується практикою – урядові організації багатьох країн використовують їх як основний засіб.

На сьогоднішній день розроблені й застосовуються відкриті формати практично для всіх класів завдань, розв'язуваних ППП, починаючи від офісних додатків до мультимедійних даних і 3D-графіки.

Додатки за запитом

Поширення веб-сервісів на основі відкритих стандартів веде до ситуації, коли замість запуску певних програм корпоративні користувачі зможуть одержати доступ до будь-яких прикладних засобів, необхідним у цей момент, просто підключившись до Інтернет. У такому контексті додатка можуть бути представлені як вільно й безкоштовно, так і платно, по підписці, залежно від обсягу споживання.

Сполучення широкополосного інтернету із платформонезалежними додатками (написаними, приміром, мовою Java) у деяких областях уже зробили модель комунальних послуг в області ІТ реальністю. Наприклад, Salesforce.com за помірну місячну плату пропонує в Інтернеті додатка для керування відносинами із клієнтами (CRM). Користувачам, кількість яких уже становить близько 100 тис., не потрібно встановлювати або підтримувати в себе складні пакети CRM. Їм досить тільки запустити браузер і підключитися до серверу і послуг. У свою чергу вільний доступ до офісних додатків представляє раніше згаданий сервіс Google Docs, число користувачів якого постійно росте.

Розробка структурної схеми

Розглянемо структурну схему системи, зображену на рисунку 1. В архітектурі можна виділити наступні складові:

- систему управління РОС (СУРОС);
- набір обчислювальних клієнтів РОС;
- систему зберігання даних;
- засоби доступу користувачів до РППП.

Процес виконання (інтерпретації) СРЗ у РОС являє собою обчислювальний процес, що включає трансляцію цієї схеми у внутрішнє подання і її покрокову інтерпретацію. Інтерпретація операторів виклику модулів вимагає виконання відповідних модулів у вузлах РОС.

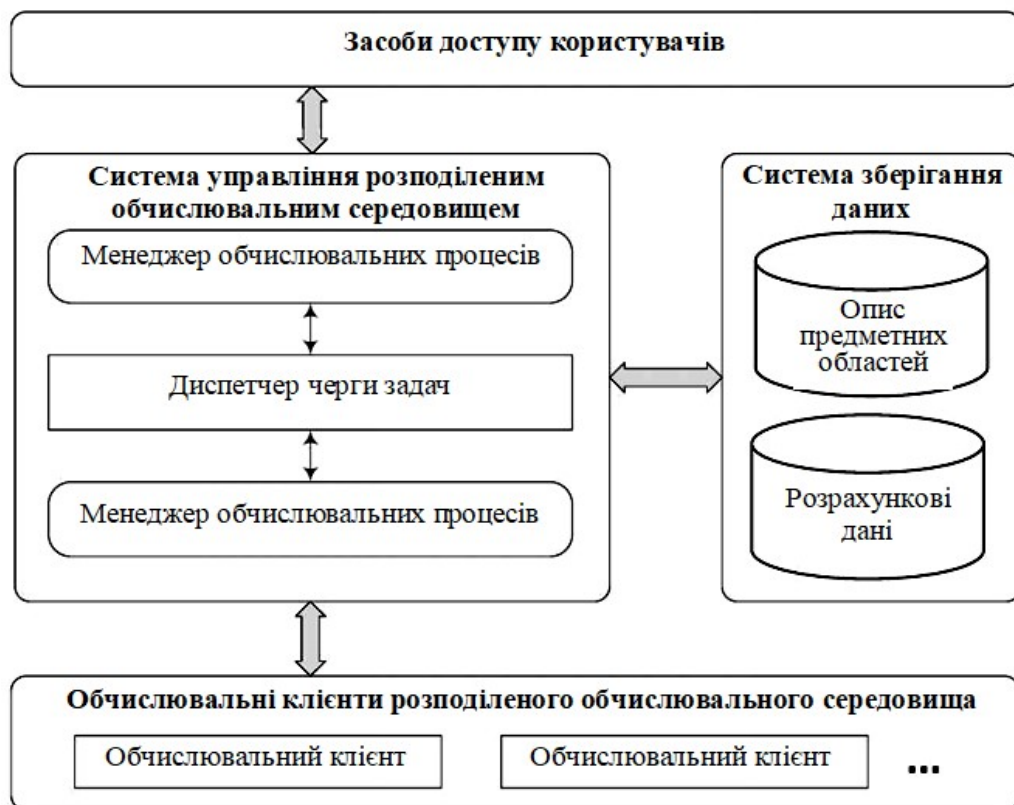


Рисунок 1 – Структурна схема системи

Яке майбутнє упаковки програм?

У сучасному світі 99% організацій почали думати про сучасне управління та цифрову трансформацію. Ми відходимо від традиційного робочого місця? Так, ми повільно просуваємося до цифрового робочого місця.

Пакування додатків є критично важливим потоком на шляху цифрової трансформації вашої організації. Команда, яка розробляє пакети програм, має зазнати трансформації та спробувати відійти від традиційного пакування програм. Більшість організацій переходять від традиційних до сучасних технологій керування пристроями.

Кожна організація повинна негайно почати трансформацію сучасного управління пристроями

У рамках шляху до цифрової трансформації команда з розробки пакетів додатків повинна мати можливість конвертувати традиційний пакет MSI у MSIX.

Подорож до цифрової трансформації вимагатиме значно більше зусиль щодо персоналізації, тому найближче майбутнє пакування є світлим.

Сучасний інструмент упаковки програм

Я бачив, як пакувальна команда витрачає свій час на повторювані завдання. Деякі приклади: підготовка віртуальних машин до тестування пакетів і застосування однакових параметрів налаштування до всіх пакетів.

Сучасні інструменти пакування можуть допомогти вам автоматизувати багато повторюваних завдань пакування.

Я вважаю, що будь-який сучасний інструмент для упаковки повинен допомогти організаціям здійснити цифрову трансформацію. Давайте перевіримо, чи має студія розумної упаковки такі характеристики.

- Машинне навчання та штучний інтелект.
- Наступний рівень автоматизації (унікайте повторюваних завдань).
- Сучасний інструмент упаковки додатків для швидшого створення пакета.
- Розумна інтеграція з віртуальною інфраструктурою для тестування пакетів.
- Розширені параметри усунення несправностей.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів реалізації інструментальних засобів організації пакетів прикладних програм. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем реалізації інструментальних засобів організації пакетів прикладних програм; Досліджена система реалізації інструментальних засобів організації пакетів прикладних програм; На основі отриманих результатів досліджень створена програмна реалізація системи реалізації інструментальних засобів організації пакетів прикладних програм. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання реалізації інструментальних засобів організації пакетів прикладних програм. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
2. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.
3. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.
4. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced

- Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
5. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 3(73), С. 155-166.
 6. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». Кібербезпека: освіта, наука, техніка. 2024. №3(23), С. 111-131.
 7. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 2(72), С. 170-178.
 8. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.
 9. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.
 10. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
 11. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
 12. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
 13. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
 14. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
 15. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.
 16. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018
 17. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for Dom Xss vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Електронний Журнал]. Georgia. Tbilisi: SCSA – 2018.
 18. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.
 19. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215
 20. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the dom xss vulnerability. International Conference «information technologies, systems and networks ITSN-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.
 21. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібербезпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA - 2017.
 22. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). - Полтава: ПолтНТУ. - 2017. - С. 112-115.
 23. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів. Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.
 24. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). - Полтава: ПолтНТУ. - 2016. - С. 98-103.