

Похиленко О.А.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Баклан І.В.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

СУПРОВОДЖЕННЯ ГІБРИДНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ, ЯКІ ЕВОЛЮЦІОНУЮТЬ

У статті розглянуті гібридні інтелектуальні системи (ГІС), інтелектуальні системи, що еволюціонують, (ЕІС), а також гібридні інтелектуальні системи, які еволюціонують, (ЕГІС) – тип інтелектуальних систем, що поєднує переваги ГІС та ЕІС. Незважаючи на підвищений інтерес до інтелектуальних систем, а також перспективності їхнього використання, виявлено, що супроводженню ЕГІС приділено недостатньо уваги. Досліджено загальний процес супроводження програмного забезпечення та виділено його фази. Розглянуті фази були застосовані щодо супроводження ЕГІС. Було розглянуто різні методи супроводження інтелектуальних систем і запропоновано їхнє використання для супроводження ЕГІС. Розглянуто вплив відповідних методів на фази супроводження ЕГІС. Запропоновано поєднання методів для покращення супроводження ЕГІС. Також було виявлено, що ЕГІС доцільно створювати у вигляді багатоагентних систем. Було розглянуто можливість створення агентів супроводження, які дозволяють автоматично вносити зміни до системи з метою запобігання її деградації. Запропоновано використання агентів супроводження як допоміжного елемента для покращення супроводження ЕГІС. Виявлено перспективність розробки багатоагентних ЕГІС із наявністю агентів супроводження та застосуванням метаданих для створення агентів із самоописом із метою покращення супроводження ЕГІС. Припущено, що додавання метаданих до агентів також може бути використано для створення довідкової системи, а це може підвищити зручність користування ЕГІС. Також було проаналізовано архітектуру ГІС та ЕІС, запропоновано способи їхнього перетворення на ЕГІС. Розглянуто деякі сучасні архітектури інтелектуальних систем, виділено окремі їхні модулі. Доведено належність розглянутих систем до ЕГІС, а отже, і можливість застосування до них запропонованих методів і підходів щодо супроводження. Припущено, що представлення знань і супроводження баз знань в ЕГІС потребує окремого розгляду. Виявлено, що супроводження ЕГІС заслуговує на глибше дослідження.

Ключові слова: архітектура програмного забезпечення, багатоагентна система, гібридна інтелектуальна система, представлення знань, обчислювальний інтелект, супроводження програмного забезпечення.

Постановка проблеми. Останнім часом спостерігається підвищений інтерес до розробки та покращення архітектур інтелектуальних систем (ІС). Одним зі стимулів є необхідність створення ІС, здатних вирішувати завдання у предметних галузях, що важко формалізуються, з неможливістю отримати повну й абсолютно достовірну інформацію. У таких випадках для вирішення завдань можуть бути використані гібридні інтелектуальні системи (ГІС), які використовують кілька різних методів обчислювального інтелекту.

Враховуючи величезну кількість інформації, котру треба обробляти у сучасному світі, іншим стимулом для покращення ІС можна назвати необхідність обробки потокової інформації. Нині для цього ведеться розробка інтелектуальних систем,

які еволюціонують (ЕІС). Такі системи оновлюють свої параметри і навіть структуру на основі потоків даних.

Поєднання підходів, що застосовуються у ГІС та ЕІС, також може бути використане для створення гібридних інтелектуальних систем, які еволюціонують (ЕГІС). Ці системи є найбільш перспективними через високу адаптивність і можливість вирішувати складні завдання, такі як прийняття рішень на основі потоків даних в умовах обмеженої та нечіткої інформації.

Поєднання методів і підходів ГІС та ЕІС призводить до того, що ЕГІС є системами, які важко супроводжувати, оскільки вони складаються з багатьох компонентів і використовують достатньо складні моделі, параметри яких можуть змінюватися із часом.

Враховуючи складність і багатокомпонентність ЕГПС, а також їхню відмінність від класичних ІС, можна зробити припущення про необхідність та актуальність розгляду методів їхнього супроводження.

Аналіз останніх досліджень і публікацій. У роботі [1] був проведений аналіз відмінностей різних класів архітектур ГПС, а також переваги та недоліки деяких підходів щодо побудови ГПС, але питанню супроводження ГПС приділено недостатньо уваги. У [2] більш детально розглянуті різні методи та моделі, що використовуються для побудови ГПС, деякі з яких підтримують можливість навчання й адаптації під час роботи системи, а отже, можуть бути використані для побудови ЕГПС, але супроводження таких систем було розглянуто недостатньо детально.

Окремої уваги заслуговує використання багатоагентної технології для побудови ІС. У [3] розглядається побудова ГПС на основі багатоагентних систем, що призводить до спрощення розробки складних ГПС. У статті [4] пропонується підхід, котрий дозволяє надати агентам таких систем властивість до самоопису за рахунок додавання анотацій, що у теорії також може покращити супроводжуваність багатоагентних ЕГПС. У роботі [5] пропонується метод супроводження та запобігання деградування, який може бути застосований до багатоагентних систем і в теорії багатоагентних ЕГПС.

У роботі [6] розглядається проблема супроводження ЕІС і згадується необхідність використання спеціалізованих засобів супроводження, що пристосовані до змінної природи цих систем. ЕІС та ЕГПС також розглядалися у працях [7–8], але їхньому супроводженню приділено недостатньо уваги.

Слід зазначити, що, незважаючи на наявність досліджень і публікацій, у яких розглядаються ГПС, ЕІС та ЕГПС, їхньому супроводженню (особливо ЕГПС) приділяється недостатньо уваги, або воно взагалі не розглядається.

Постановка завдання. Виходячи з аналізу публікацій, можна сказати, що супроводження ЕГПС потребує додаткового дослідження. Отже, метою цієї роботи є проведення аналізу й оцінки можливостей застосування методів супроводження щодо ЕГПС. Для цього були визначені такі завдання дослідження: розглянути різні методи супроводження ІС, архітектури ГПС та ЕІС, після чого проаналізувати переваги та недоліки розглянутих методів супроводження і виявити, які з цих методів можна використовувати для ЕГПС.

Виклад основного матеріалу дослідження. Відповідно до стандарту [9] здатність до супро-

дження є одним із показників якості програмного забезпечення, до якого належать модульність, а також здатності до повторного використання, аналізу, модифікації та тестування. У статті супроводження ЕГПС буде розглядатися саме з позиції цих показників. Використовуючи термінологію зі стандарту [10], акцент буде робитися на коригувальному супроводженні та зручності супроводження з погляду внесення модифікацій, що можуть включати покращення й адаптацію програмного забезпечення до змін в оточенні чи специфікаціях.

Супроводження. Враховуючи поставлені завдання, доцільно було б розпочати дослідження із загальних технік супроводження програмного забезпечення. В [11, с. 3] наводиться процес супроводження програмного забезпечення, зображений на рис. 1. Виділяються 4 основні фази:

1) аналіз портфелю та визначення стратегії – включає оцінку поточного стану системи, встановлення вимог і цілей щодо супроводження, а також розробку відповідної стратегії та заходів;

2) моделювання й аналіз системи – передбачає представлення системи у певному формальному вигляді з метою виявлення важливої інформації щодо структури системи, залежностей і характеристик;

3) трансформація артефакту – має на меті застосування різних методів для досягнення вимог і цілей, визначених у фазі 1;

4) оцінка системи – виконується для виявлення досягнення вимог і цілей за рахунок обчислення показників якості системи після її трансформації та порівняння їх із попередніми значеннями.

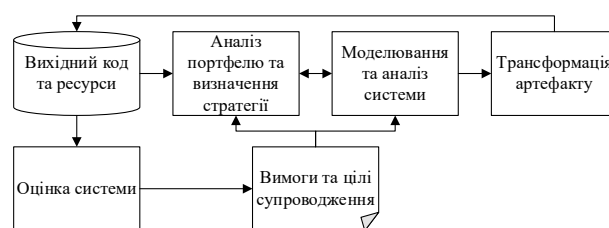


Рис. 1. Процес супроводження програмного забезпечення з [11, с. 3]

Розглянуті фази виконуються ітераційно, але процес передбачає можливість повертання до попередніх фаз: наприклад, результати фази 2 можуть призвести до необхідності перегляду та коригування стратегії фази 1. Моделювання й аналіз систем ЕІС та ЕГПС може бути ускладнений через їхню можливість змінювати власну структуру та параметри під час роботи. Також при супроводженні цих систем потрібно брати до уваги можливість деградації, для виявлення якої потрібно періодично про-

водити оцінку поточного стану системи. Тоді при виявленні падіння показників і наявності стратегії їхнього підвищення необхідно вжити відповідні заходи щодо трансформації системи.

Цікавим є те, що в [11, с. 9] згадується, що методи трансформації з фази 3 можуть бути також напівавтоматичними й автоматичними, що особливо важливо для ЕІС та ЕГІС. Отже, наявність автоматичних засобів запобігання деградації у цих системах у певних випадках зможе спростити їхнє супроводження. Саме такий метод розглядався у [5, с. 386], де для запобігання деградації у багатоагентних ІС пропонувалося використання окремих агентів (агентів супроводження), що стежать за показниками якості ІС та виконують певні дії у разі їх падіння. Такий підхід не позбавляє від необхідності ручного виконання фаз 1–2, оскільки для створення агентів супроводження спочатку необхідно реалізувати відповідні алгоритми, для чого треба визначити, за якими показниками потрібно стежити та власне яким чином покращувати якість цих показників. Враховуючи обмеженість агентів супроводження та практичну неможливість передбачення усіх можливих ситуацій деградування у складних ЕГІС, можна стверджувати, що застосування автоматичних методів супроводження повністю не позбавить від необхідності використання відповідних ручних методів.

Враховуючи те, що ГІС та ЕГІС є системами, які зазвичай складаються з багатьох компонентів, також доцільно розглянути окремі методи супроводження та повторного використання компонентів. Супроводження таких складних систем можна було би спростити за рахунок використання методології, заснованої на інформаційному пошуку й автоматичної каталогізації, що розглядалася у [12]. Запропонована у [12, с. 196] система, яка використовує цю методологію, призначена для застосування з уже написаними програмними компонентами (власне вихідним кодом), але потенційно також може бути модифікована для використання разом із фреймворком, запропонованим у [4], що використовує анотації для опису агентів у багатоагентних системах. Таким чином, теоретично можливо адаптувати методологію з роботи [12] для її використання у багатоагентних ІС під час роботи системи. Це може дозволити спростити аналіз поточного стану системи та виявлення певної додаткової інформації за рахунок використання інформаційного пошуку до активних агентів системи. Таким чином, поєднання методів, запропонованих у [4; 12] може призвести до покращення фаз 1–2 супроводження ЕГІС.

Опис компонентів складних систем із метою подальшого покращення їхнього супроводження, а саме здатності до аналізу та тестування, також розглядається у роботі [13]. Запропонована методологія передбачає додавання метаданих і застосування методів Семантичної Павутини, що включає використання RDF та OWL для представлення інформації, а також SPARQL для її вилучення. Враховуючи вже розглянуті дослідження, перспективним було би поєднання цієї методології із фреймворком з [4]. Таке поєднання дозволить досягти стандартизованого опису агентів, що може бути застосовано у багатоагентній ЕГІС.

Методи, запропоновані у [4; 13] потребують наявності метаданих щодо можливостей системи. Таким чином, за відсутності відповідних метаданих їхнє використання буде неможливим. У цьому разі можна або вручну додати потрібні метадані, або автоматично згенерувати їх. Один із підходів такої автоматичної генерації розписаний у [14]. Отже, поєднання запропонованих підходів у [4; 12–14] із погляду супроводження призведе до покращення фаз 1–2. Цікавим є те, що додавання певних метаданих і їхня доступність користувачам системи може підвищити зручність користування, створюючи довідкову систему. Таким чином, за певних модифікацій запропонована методологія може одночасно підвищити зручності супроводження та користування ЕГІС.

Архітектури. Розглянувши деякі методи супроводження, потрібно проаналізувати можливість їхнього застосування до архітектур ЕГІС. Оскільки у більшості публікацій ЕГІС не виокремлюються, виконаємо аналіз на основі публікацій щодо архітектур ГІС та ЕІС, враховуючи те, що фактично ЕГІС як категорія знаходиться на перетині ГІС та ЕІС. Отже, здебільшого для отримання ЕГІС необхідно або додати можливість обробки потокової інформації та зміни власних параметрів до ГІС, або використати декілька різних методів обчислювального інтелекту в ЕІС. Таким чином, далі будуть розглянуті архітектури різних ІС із доведенням того, що вони належать до ЕГІС, або зі способами їхньої модифікації для того, щоб у результаті отримати ЕГІС.

Слід розпочати з досліджень [1; 2; 15], де наведені загальні архітектури ГІС. Отже, відповідно до [1, с. 261] виділяють такі основні архітектури ГІС: уніфіковану нейронну, трансформаційну та гібридну модульну. Також окремо розглядається автономний тип архітектури, у якому модулі є незалежними один від одного. У контексті нашої роботи найбільший інтерес становить гібридна

модульна архітектура, котра використовує нейронний і символний модулі. В [1, с. 262] згадується, що виділяють три типи цієї архітектури: слабо зв'язану, сильно зв'язану та повністю інтегровану. Відмінності між цими типами полягають у взаємодії між модулями цих ГІС: у слабо зв'язаній модулі розділені, а зв'язок між ними є одностороннім; у сильно зв'язаній модулі взаємодіють із використанням спільних внутрішніх структур даних, причому виконується двонаправлений обмін знаннями між ними; у повністю інтегрованої відсутня зовнішня відмінність між модулями, а модулі мають єдиний інтерфейс. Для того, щоб розглянуті архітектури можна було б вважати такими, які належать до архітектур ЕГІС, до них потрібно ввести модулі, що можуть впливати на параметри або структуру системи. Розглядаючи [2, с. 1–3], можна виявити, що для таких цілей пропонується використання алгоритмів глобальної оптимізації та генетичних алгоритмів. Отже, побудова ЕГІС буде можливою за рахунок введення відповідних модулів до сильно зв'язаної або повністю інтегрованої гібридної модульної архітектури. Слабо зв'язана архітектура буде гірше підходити для ЕГІС через односторонність зв'язків у ній.

Відповідно до [15, с. 710] інколи окремо ще виділяють ієрархічну архітектуру, у якій модулі утворюють певну ієрархію і взаємодія відбувається лише між модулями сусідніх рівнів ієрархії. У [2, с. 3] також згадується, що ієрархічна гібридна архітектура може використовувати модуль із генетичними алгоритмами як перший модуль найвищого рівня ієрархії, який може бути використаний для оптимізації топології. Це свідчить про те, що такий різновид належатиме до ЕГІС через можливість системи змінювати власну структуру та параметри.

У [2, с. 4] та [15, с. 711] також згадується, що для більш простого розділення на модулі та кращого розуміння (а отже, і здатності до аналізу) ГІС доцільним є їхнє проектування у вигляді багатоагентних систем. Це також підтверджується тим, що у [3; 5] розглядалися саме багатоагентні ГІС. Таким чином, враховуючи процес супроводження, зображений на рис. 1, та фази супроводження, розглянуті раніше, можна зробити висновок, що побудова ЕГІС у вигляді багатоагентних систем спростить їхнє моделювання й аналіз (фазу 2 супроводження). Також варто згадати, що поєднання запропонованих підходів у [4, 12–14] теж призведе до покращення фаз 1–2, а підхід, розглянутий у [5], дозволить спростити фазу 3 за раху-

нок автоматичного методу трансформації системи агентом супроводження при виявленні падіння певних її параметрів на фазі 4 оцінки системи.

Модулі архітектур. Для кращого розуміння архітектур ЕГІС необхідно більш детально розглянути деякі сучасні приклади. Для цього було вирішено обрати статті [16–19] та довести належність розглянутих систем до ЕГІС.

У [16] розглядається підхід до побудови інтелектуальних інформаційних систем (ІС) на основі онтологій та об'єктно-орієнтованих семантичних мереж (фреймових мереж), описана архітектура такої системи та моделі представлення знань. Розглянута ІС складається із семантичного ядра та підсистем, що відповідають: за розробку, верифікацію та супроводження системи знань, за розробку та розвиток контенту, а також за представлення знань користувачу та іншим інтелектуальним агентам. Було запропоновано розділення семантичного ядра на дві частини: менеджер мережі, який відповідає за роботу з мережею на рівні даних, і семантичний модуль, що забезпечує роботу з мережею на рівні системи знань ІС. Для реалізації моделі даних є можливим використання як і реляційних СУБД, так і RDF-сховищ. При розробці онтології ІС використовувалося розділення на три базові онтології: діяльності, базових задач і предметного знання. Для роботи з онтологією ІС система також містить модулі поповнення контенту, які можуть бути використанні для ручного й автоматичного занесення знань, модулі візуалізації, аналізу та верифікації, а також модулі, що відповідають за навігацію та пошук інформації. Враховуючи наявність різних методів обчислювального інтелекту, запропонована у [16] система належить до ГІС. Через можливість автоматичного занесення знань ця ГІС може бути перетворена на ЕГІС, якщо система буде відповідним чином реагувати на нові знання, змінюючи власні параметри чи навіть структуру.

У [17] розглядається створення ІС із позиції інтелектуального інтерфейсу користувача та використання клієнт-серверної архітектури. Пропонується реалізація REST API сервісу, що оброблює запити від клієнтських застосунків. Цікавою особливістю є наявність декількох баєсових мереж на віддаленому сервері – окремо виділяється колективна мережа та персональні мережі. Таким чином, ІС містить ще й центр автентифікації, а база знань складається із загальної та персональних. Такий підхід надає можливість адаптації запропонованого рішення під потреби користувачів. Власне адаптація під користувача й обробка потокової

інформації свідчить про те, що система належить до ЕІС. Систему можна вдосконалити за рахунок додавання нових модулів, перетворивши її на ЕГІС.

У [18] пропонується використання гібридної інтелектуальної інформаційної системи (ГІС) як основи когнітивної архітектури. Використовуючи вже введenu термінологію, можна сказати, що ГІС може розглядатися як ГІС, а отже, усе вищесказане також може бути застосовано і для ГІС. У роботі пропонується розділення системи на модулі підсвідомості та свідомості, де модуль підсвідомості використовує нейронні мережі, нечітку логіку та нейро-нечіткі методи, а модуль свідомості – продукційні правила та модель на основі онтології. Для взаємодії з іншими подібними ГІС пропонується додавання третього модуля – модуля взаємодії. Взаємодія розглянутої ГІС із середовищем виконується через модуль підсвідомості, що отримує та перетворює дані, які надійшли ззовні, та передає їх до модуля свідомості. Після цього виконується логічна обробка даних і їхнє повернення до попереднього модуля. У якості моделі даних запропоноване використання моделі на основі метаграфу. Для трансформації даних метаграфу під час роботи ГІС було запропоновано використовувати агентів чотирьох видів: функціональних, що мають вхідні та вихідні параметри у вигляді метаграфів; заснованих на правилах, які застосовують продукційні правила до заданого стану; контейнерних, що складаються з інших агентів; і динамічних, які можуть змінювати структуру інших агентів. Розглянуті метаграфові агенти також задаються фрагментами метаграфу, що надає агентам можливість змінювати структуру інших агентів. У [18] також було розглянуто застосування описаної ГІС у циклі сприйняття-дія, що застосовується у когнітивних архітектурах. На етапі сприйняття ГІС отримує сигнали (візуальні, звукові чи інші) із зовнішнього середовища, після чого модуль підсвідомості розпізнає елементи онтологій і передає їх до модуля свідомості. Далі модуль свідомості за допомогою метаграфових агентів розпізнає ситуацію та формує необхідні дії, виходячи з відомих системі ситуацій. Дії передаються модулю підсвідомості, який виконує їх, генеруючи певні сигнали. У разі потреби модуль свідомості також може відправляти й отримувати дані у вигляді метаграфів від модуля взаємодії. У розглянутій статті також було припущено, що елементи запропонованої ГІС можна використати не тільки для реалізації циклу сприйняття-дія, але і для когнітивних архітектур загалом як будівельні блоки. Отже, запропонована

у [18] система є багатоагентною ЕГІС, оскільки у ній наявні агенти, що можуть змінювати параметри та структуру системи з метою адаптації під зміни у зовнішньому середовищі.

У [19] розглядається архітектура персонального когнітивного асистента. Особливістю запропонованого рішення є наявність декількох картин світу: власної картини світу асистента та картини світу користувача, що будується асистентом для його користувачів. Запропонована картина світу є знаковою і становить множину каузальних матриць і семантичних мереж, що використовуються для представлення як декларативних, так і процедурних знань. Така модель потребує попереднього навчання та налаштування сценаріїв, але також передбачає можливість автоматичного чи автоматизованого поповнення картини світу. Власна картина світу агенту може бути або статичною, або такою, що еволюціонує під час роботи. Враховуючи можливість використання різних методів обчислювального інтелекту, які можуть бути використані при розробці такого асистента, запропоновану у [19] систему можна легко перетворити на ГІС. Розглянута система також належить до ЕІС через можливість автоматичного поповнення картини світу, її еволюції й адаптації системи під користувача. Отже, можна стверджувати, що система буде належати до ЕГІС за виконання цих вимог.

У розглянутих системах використовується представлення знань, розгляд якого потребує окремого дослідження. Варто сказати, що супроводження ЕГІС буде включати не лише супроводження відповідного програмного забезпечення, а й інформаційного, тобто супроводження бази знань і контроль актуальності цих знань. У [20] згадується, що для представлення знань можуть використовуватися різні моделі, серед яких є ті, що побудовані на основі каскадної моделі, та логічні продукційні моделі. Отже, доцільно буде розглянути окремо підходи щодо супроводження ЕГІС із позиції представлення знань у цих системах.

Слід зауважити, що у [2, с. 2–3] серед загальних модулів ГІС також виділяються модулі, що включають штучні нейронні мережі, нечітку логіку та навіть експертні системи, які розглядаються у [21]. Враховуючи те, що розглянуті системи або належать до ЕГІС, або можуть бути відносно просто перетворені на ЕГІС за рахунок введення додаткових модулів, можна стверджувати, що розглянуті методи супроводження будуть підходити для використання із цими системами. Особливо це стосується розглянутих методів супроводження багатоагентних систем.

Висновки. Отже, був розглянутий загальний процес супроводження програмного забезпечення та виділені його фази. Аналіз літератури показав, що супроводження ЕГІС нині розглянуто недостатньо добре. Було виявлено, що ЕГІС можуть бути побудовані у вигляді багатоагентних систем. Знайдено підходи, які можуть покращити деякі фази супроводження програмного забезпечення ЕГІС. Було розглянуто створення агентів супроводження, що можуть автоматично виконувати

коригування параметрів під час роботи ЕГІС. Також проаналізовано найпопулярніші архітектури ГІС та ЕІС, виявлено способи їхнього перетворення на ЕГІС. Було виділено окремі модулі декількох сучасних архітектур ІС, доведено їхню належність до ЕГІС або способи їх перетворення на ЕГІС. Таким чином, можна стверджувати, що розробка та покращення ЕГІС є перспективним та актуальним напрямком, а супроводження ЕГІС заслуговує на подальше дослідження.

Список літератури:

1. Крапивный Ю.Н., Кривонос А.О. Архитектуры гибридных интеллектуальных систем. *Електротехнічні та комп'ютерні системи*. 2016. № 22. С. 260–266. URL: http://nbuv.gov.ua/UJRN/etks_2016_22_45.
2. Abraham A., Nath B. Hybrid intelligent systems design: A review of a decade of research. *IEEE Transactions on Systems, Man & Cybernetics (Part-C)*. August, 2000. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.469&rep=rep1&type=pdf>.
3. Zhang Z., Zhang C. Building agent-based hybrid intelligent systems: A case study. *Web Intelligence and Agent Systems*. 2007. Vol. 5. № 3. P. 255–271.
4. Goerz G., Ludwig B., Reiß P., Schiemann B., Seutter T. Self-describing Agents. *Multikonferenz Wirtschaftsinformatik : Proceedings of MKWI 2008, München, February 26–28, 2008*.
5. Ghrieb N., Mokhati F., Guerram T. Maintaining Organizational Multi-agent Systems: A Reorganization-based Preventive Approach. *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*. February 4–6, 2021. Vol. 1. P. 384–389. DOI: <https://doi.org/10.5220/0010314803840389>.
6. Ramamoorthy C.V., Cooke D.E. The Correspondence Between Methods of Artificial Intelligence and the Production and Maintenance of Evolutionary Software. *Proceedings of Third International Conference on Tools for Artificial Intelligence*. November 10–13, 1991. P. 114–118. DOI: <https://doi.org/10.1109/TAI.1991.167083>.
7. Lemos A., Caminhas W., Gomide F. Evolving Intelligent Systems: Methods, Algorithms and Applications. *Emerging Paradigms in Machine Learning. Smart Innovation, Systems and Technologies*. 2013. Vol. 13. P. 117–159. DOI: https://doi.org/10.1007/978-3-642-28699-5_6.
8. Abraham A. Hybrid Intelligent Systems: Evolving Intelligence in Hierarchical Layers. *Do Smart Adaptive Systems Exist? Studies in Fuzziness and Soft Computing*. 2006. Vol. 173. P. 159–179. DOI: https://doi.org/10.1007/3-540-32374-0_8.
9. ДСТУ ISO/IEC 25010:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів (ISO/IEC 25010:2011, IDT).
10. ДСТУ ISO/IEC 14764:2014 Інженерія програмного забезпечення. Процеси життєвого циклу програмного забезпечення. Технічне обслуговування (ISO/IEC 14764:2006, IDT).
11. Kontogiannis K. Techniques for Software Maintenance. *Encyclopedia of Software Engineering*. 2011.
12. Pighin M. A New Methodology for Component Reuse and Maintenance. *Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR '01)*. Lisbon, Portugal, Mar. 14–16, 2001. P. 196–199.
13. Hyland-Wood D., Carrington D., Kaplan S. Towards a software maintenance methodology using Semantic Web techniques and paradigmatic documentation modelling. *IET Software*. 2008. Vol. 2. No. 4. P. 337–347. DOI: <https://doi.org/10.1049/iet-sen:20070092>.
14. Abid N.J., Dragan N., Collard M.L., Maletic J.I. Using stereotypes in the automatic generation of natural language summaries for C++ methods. *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 2015, P. 561–565. DOI: <https://doi.org/10.1109/ICSM.2015.7332514>.
15. Jacobsen H.-A. A generic architecture for hybrid intelligent systems. *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228)*. 1998. Vol. 1. P. 709–714. DOI: <https://doi.org/10.1109/FUZZY.1998.687575>.
16. Загоруйко Ю.А. Подход к построению интеллектуальных информационных систем на основе семантических сетей. *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2011)*. Минск, февраль 10–12, 2011. С. 15–20.
17. Левус Є.В., Гучок Ю.О., Нитребич О.О. Архітектурне рішення для побудови інтелектуальних інтерфейсів користувача. *Науковий вісник НЛТУ України*. 2018. Т. 28, № 8. С. 155–160. DOI: <https://doi.org/10.15421/40280831>.

18. Chernenkiy V., Gapanyuk Y., Terekhov V., Revunkov G., Kaganov Y. The hybrid intelligent information system approach as the basis for cognitive architecture. *Procedia Computer Science*. 2018. Vol. 145. P. 143–152. DOI: <https://doi.org/10.1016/j.procs.2018.11.022>.

19. Смирнов И.В., Панов А.И., Скрынник А.А., Чистова Е.В. Персональный когнитивный ассистент: концепция и принципы работы. *Информатика и её применения*. 2019. Т. 13. № 3. С. 105–113. DOI: <https://doi.org/10.14357/19922264190315>.

20. Польшкалин В.Я., Баклан И.В. и др. Информационное обеспечение интегрированных АСУ ГПС. Т. 2. Москва : Машиностроение, 1989. 888 с.

21. Баклан И.В. Експертні системи. Курс лекцій : навчальний посібник. Київ : НАУ. 2012. 132 с.

Pokhylenko O.A., Baklan I.V. MAINTENANCE OF EVOLVING HYBRID INTELLIGENT SYSTEMS

The article considers hybrid intelligent systems (HIS), evolving intelligent systems (EIS), and evolving hybrid intelligent systems (EHIS) – a type of intelligent systems that combines the advantages of HIS and EIS. Despite the increased interest in intelligent systems, as well as the prospects for their use, it was found that the maintenance of EHIS is given insufficient attention. The general process of software maintenance and its phases are investigated. The considered phases were applied to the maintenance of EHIS. Various methods of maintenance of intelligent systems were considered and their use for maintenance of EHIS was suggested. The influence of appropriate methods on the phases of maintenance of EHIS is considered. A combination of methods to improve maintenance of EHIS is proposed. Also, as a result of the study, it was found that EHIS should be created in the form of multi-agent systems. The possibility of creating maintenance agents that allow to automatically make changes to the system to prevent its degradation was considered. The use of maintenance agents as an auxiliary element to improve maintenance of EHIS is proposed. Prospects for the development of multi-agent EHIS with the use of maintenance agents and the use of metadata to create self-describing agents in order to improve the maintenance of EHIS are considered. It is suggested that adding metadata to agents can also be used to create a help system, which can increase the usability of EHIS. Also in the article the architectures of HIS and EIS were analyzed, and the ways of their transformation into EHIS were offered. Some modern architectures of intelligent systems and their separate modules are considered. The belonging of the considered systems to EHIS is proved, and hence the possibility of applying the proposed methods and approaches of maintenance. It is assumed that the presentation of knowledge and maintenance of knowledge bases in EHIS requires separate consideration. As a result, it was found that the maintenance of EHIS deserves more in-depth study.

Key words: computational intelligence, hybrid intelligent system, knowledge representation, multi-agent system, software architecture, software maintenance.