

УДК 004.94

## ОНТОЛОГІЧНИЙ ПІДХІД ДО КОНСТРУЮВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА В СИСТЕМАХ ІНДУКТИВНОГО МОДЕЛЮВАННЯ

Піднебесна Г.А.

*Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України, м. Київ*

*pidnebesna@irtc.org.ua*

В роботі розглядається онтологічний підхід до побудови інтерфейсу користувача при конструюванні засобів індуктивного моделювання, заснованих на знаннях.

*Ключові слова: індуктивне моделювання, МГУА, онтологія, інтерфейс користувача*

This paper considers the ontological approach to structuring of user interface in the design of inductive modeling tools.

*Keywords: inductive modeling, GMDH, ontology, user interface.*

В работе рассматривается онтологический подход к построению пользовательского интерфейса при конструировании средств индуктивного моделирования, основанных на знаниях.

*Ключевые слова: индуктивное моделирование, МГУА, онтология, пользовательский интерфейс.*

**Вступ.** Однією з найбільш актуальних проблем сучасного суспільства є постійно зростаюча надзвичайно велика кількість інформації, яку необхідно вміти отримати, обробити, аналізувати, зберігати тощо. Основним напрямком для подолання «інформаційного переповнення» визначено перехід від зберігання та обробки даних до накопичення та обробки знань [1]. Цим обумовлено бурхливий розвиток досліджень в галузі штучного інтелекту, зокрема, в розробці «інтелектуальних» комп'ютерних систем, заснованих на знаннях.

Збільшується обсяг даних, а відповідно, обсяг задач обробки та аналізу цих даних, моделювання за ними складних процесів та явищ, прогнозування для підтримки прийняття рішень тощо. Постійно виникають та розвиваються інструментальні засоби для вирішення цих задач. Розробка інтелектуальних комп'ютерних систем – процес надзвичайно складний та коштовний. Крім розробки постійно виникають проблеми з необхідністю внесення змін в уже існуючі системи, пристосування їх до нових версій та можливостей новітніх інформаційних технологій. Тобто, основними вимогами в сучасних задачах конструювання програмних засобів залишається спрощення процесу конструювання та модифікації, а відповідно, зменшення вартості таких робіт. Це досягається переходом від процедурного програмування до ООП, декларативних описів, що дає можливість розробки мовно- та платформо-незалежних моделей; з автоматичною генерацією (повною або частковою)

коду; повторним використанням, роздільною модифікацією архітектурних компонент; поділом кожного архітектурного компонента на складові, тощо.

Інтерфейс користувача є невід'ємною частиною сучасних програмних засобів, інтелектуальних комп'ютерних систем. Тому всі зазначені вище проблеми і методи їх подолання в повній мірі відносяться до розробки інтерфейсу користувача.

## **1. Інтерфейс користувача як значуща частина програмних комплексів, заснованих на знаннях**

Інтерфейс користувача (UI- англ. User interface) являє собою сукупність засобів і методів, за допомогою яких людина (користувач) взаємодіє з комп'ютерними програмними засобами, найчастіше складними, з великою кількістю елементів.

Ці засоби та методи можуть бути різними:

- Інтерфейс командного рядка: інструкції комп'ютера даються шляхом введення з клавіатури текстових рядків (команд).
- Графічний інтерфейс користувача, коли програмні функції представляються графічними елементами екрану.
- Діалоговий інтерфейс.
- Природно-мовний інтерфейс: користувач «розмовляє» з програмою рідною йому мовою.
- Тактильний інтерфейс (джойстик) тощо.

Інтерфейс користувача часто розуміють тільки як зовнішній вигляд програми, картинку, яка відображається на екрані. Однак, насправді таке розуміння є занадто вузьким. В дійсності інтерфейс користувача об'єднує всі елементи і компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням. До них відносяться:

- сукупність завдань користувача, які він вирішує за допомогою системи;
- елементи управління системою;
- навігація між блоками системи;
- візуальний (і не тільки) дизайн екранів програми;
- засоби відображення інформації;
- пристрої та технології вводу / виводу даних;
- порядок використання програми і документація на неї тощо.

Користувачі оцінюють програмний засіб не тільки за його функціональність, а й в цілому, враховуючи зручність інтерфейсу, ергономічність, наскільки він відповідає їхнім вимогам. Вимоги можуть бути різноманітними, визначатись особистими уподобаннями, виконуваними задачами, особливостями представлення інформації в різних предметних галузях, умовами та середовищем використання, рівнем підготовленості користувача тощо.

Інтерфейс користувача в системах, заснованих на знаннях, повинен допомагати користувачу будь-якого рівня підготовленості пройти всі етапи розв'язання задачі. Зокрема, в задачах моделювання за статистичними даними – від попереднього аналізу даних до оцінки адекватності побудованої моделі. Такий інтерфейс має бути «інтелектуальним».

За означенням Д.А. Поспелова [2], інтелектуальний інтерфейс користувача повинен реалізовувати функції спілкування, автоматичної побудови програми (алгоритму, сценарію), обґрунтування та навчання.

Такий інтерфейс користувача повинен бути орієнтований на користувачів різного рівня підготовленості (від новачка до експерта). Причому поради системи не повинні обмежувати творчі можливості користувача, але мають позбавляти його від помилок або непродуктивного витрачання сил.

Існуючі програмні системи, які дозволяють розв'язувати задачі моделювання на основі даних спостережень за допомогою різних методів, в тому числі МГУА, як правило поділяються на два типи: ті, що розроблені за принципом «чорного ящика», коли користувач не має можливості впливати на процес моделювання; і розраховані на спеціалістів, коли на кожному кроці необхідно задавати параметри моделювання. [3, 4]

Часто на практиці користувачі мають різні рівні підготовки:

- початківець - має дані, нічого не знає про методи їх обробки;
- досвідчений користувач - знайомий з такими задачами, знає методи розв'язання;
- експерт - знає задачу, методи розв'язання та конкретні програмні засоби.

Доцільно організувати інтерфейс і роботу системи так, щоб забезпечити можливість діалогу відповідно до кваліфікації чи задач користувача. Наприклад:

- Автоматичний режим – розраховано на непідготовленого користувача, який передає вибір системі або у випадку, коли вибір на даному етапі для користувача очевидний чи несуттєвий.
- Меню доцільних варіантів вибору – наявний перелік варіантів, що найчастіше зустрічаються (використовуються) для даного типу задач. Такий режим розраховано на досвідченого користувача, який знайомий з представленими методами або вже має досвід роботи з системою.
- Повне меню – представляє всі можливі варіанти вибору рішень на конкретному етапі (кроці) моделювання. Розраховано на користувача-експерта, що розв'язує специфічні задачі, добре знається на представлених методах та можливостях системи.

У такий інтерфейс закладено можливість на підставі певного обсягу інформації про розв'язання подібних задач сформулювати різні типові сценарії.

На кожному кроці моделювання (побудови алгоритму) перевіряється несуперечливість прийнятого користувачем рішення. При цьому передбачається блок пояснень рішень для автоматичного режиму, пропонуються рекомендації, наприклад, змінити алгоритм чи його параметри. Тоді можна працювати з іншими методами і покращувати результат моделювання.

## **2. Принципи побудови інтерфейсу користувача на основі онтологій**

Наведені фактори вимагають побудови інтелектуальних інтерфейсів користувача з різними формами представлення інформації, підтримки різних типів діалогів, для різних платформ.

Ускладнення функціональності програмних систем обумовило появу спеціальних інструментальних засобів підтримки розробки програмних комплексів та зокрема інтерфейсу користувача. В сучасних дослідженнях в галузі проектування інтерфейсу користувача можна виділити два основних напрямки: дизайнерський та моделеорієнтований:

В рамках дизайнерського напрямку розроблено методи проектування візуального представлення ІК (WIMP-інтерфейси «window, icon, menu, pointing device») та автоматичного кодування на якусь з мов програмування (Pascal, C#, C++, Java...). Це реалізується з допомогою *розробників інтерфейсу* (Interface Builders), які дають можливість конструювання тільки компонент візуалізації за допомогою існуючих бібліотек елементів інтерфейсів, використовуючи технологію візуального програмування в межах інтегрованих середовищ програмування.

В рамках моделеорієнтованого напрямку існують засоби, які підтримують роздільну реалізацію та модифікацію WIMP-інтерфейсу та прикладної програми, та мають засоби декларативного опису моделі зв'язку між інтерфейсом та прикладною програмою. Це так звані *системи управління інтерфейсу користувача* (User Interface Management Systems).

Наступним рівнем є *моделеорієнтовані засоби для розробки інтерфейсу* (Model-Based Interface Development Environment), які дозволяють створити повну специфікацію всієї моделі інтерфейсу, з розподілом на складові частини, модулі повторного використання, а також дають можливість реалізації автоматичного кодування компонентів за побудованою специфікацією інтерфейсу користувача.

Кожен з таких засобів має свій набір складових, різні компоненти описуються різними засобами та мовами, що обумовлює складність при проектуванні та подальшій модифікації.

Для подолання подібних проблем в галузях програмної інженерії поширюється застосування різних видів декларативних описів [5]. Одним з різновидів декларативного програмування є онтологічне, яке базується на використанні онтологій.

Зокрема, онтологічний підхід до конструювання інтерфейсу користувача запропоновано в [6]. Основними принципами такого підходу є наступні.

1. Розділити розробку та модифікацію інтерфейсу користувача та прикладної програми за умови виконання правил їхньої взаємодії.
2. Об'єднання в окремі компоненти інтерфейсу однорідної за змістом інформації. Наприклад, в інтерфейсі користувача можна виділити декілька основних систем понять:
  - Система понять користувача, за допомогою яких користувач здійснює взаємодію з системою;
  - Система понять розробника інтерфейсу користувача, яка має містити поняття для опису структури та засобів відображення інформації в інтерфейсі,
  - Система поняття розробника для опису сценарію діалогу;
  - Система понять розробника для здійснення взаємодії прикладної програми та інтерфейсу користувача.
3. Представити компоненти інтерфейсу користувача у вигляді декларативних моделей, сформованих на основі універсальних онтологій, що описують кожен такий компонент.
4. Автоматична генерація коду інтерфейсу за декларативними моделями, що дозволяє значно скоротити час та зусилля при розробці.

Інтерфейс користувача є тією частиною програмного комплексу, який з огляду на широке коло ймовірних користувачів найчастіше може змінюватись, враховуючи різний рівень підготовленості, широкий спектр задач та особисті уподобання. Реалізувати дружній до користувача інтерфейс, який задовольнятиме наведеним вимогам, можливо з допомогою побудови макетів майбутнього інтерфейсу, заснованих на використанні універсальних онтологій інтерфейсу користувача.

Модель будь-якого інтерфейсу користувача прикладної програми можна розглядати як сукупність наступних моделей:  $V$  – модель онтології представлення,  $S$  – модель онтології сценарію діалогу,  $L$  – модель онтології зв'язку з прикладною програмою,  $U$  – модель онтології задач користувача. Тоді формування компоненту моделі інтерфейсу зводиться до задання значень понять відповідної моделі універсальної онтології –  $\{V, S, L, U\}$ .

*Модель онтології представлення* містить в собі три різновиди моделей онтологій:

- моделі онтології графічного інтерфейсу користувача використовується для формування представлення інформації на основі екранних форм, складається з понять WIMP-інтерфейсів (Windows, Icons, Menus, Pointing devices);

- моделі онтологій статичних та динамічних сцен на площині визначають базові графічні зображення, наповнення, примітиви;

- моделі онтологій для формування текстів визначає конструкції для опису

структури та засобів побудови текстів за результатами роботи прикладної програми.

Кожна з цих моделей дозволяє будувати один з типів діалогів. Їх поєднання дозволяє отримати найбільш повне й зручне представлення, що відповідає вимогам користувача.

*Модель онтології сценарію діалогу* визначає абстрактні терміни для опису реакцій на події, сукупність дій, які виконуються при виникненні подій, джерела подій, типи переходів між вікнами, способи вибору екземплярів вікна тощо.

*Модель онтології зв'язку з прикладною програмою* визначає множину засобів і можливостей, що надаються прикладною програмою. Кожен з них описує модель взаємодії з програмою, перелік функцій, які надаються, характеризуються типом значень, множиною параметрів, їхніх типів тощо.

*Модель онтології задач користувача* містить опис предметної галузі, задач та методів їх розв'язання.

### **3. Модель онтології задач користувача для галузі індуктивного моделювання**

Модель онтології задач користувача містить множину онтологій предметної галузі, специфікує об'єкти діяльності, визначає класи термінів, групи, поняття, характеристики та області можливих значень, стану та необхідні для виконання задач зв'язків серед об'єктів. Онтологія задач необхідна для опрацювання функціональності додатків, побудови послідовності виконання дій, які вирішують конкретні типи завдань.

Наприклад, в рамках індуктивного моделювання, а саме МГУА [7], розглядається цілий спектр задач:

- побудова математичної моделі об'єкта (процесу);
- прогнозування перебігу процесу, заданого часовим рядом;
- класифікація – побудова правила віднесення об'єкта до того чи іншого класу за заданими ознаками;
- розпізнавання образів з навчанням (виявлення ефективних ознак та правила розрізнення заданих класів);
- кластеризація (самонавчання, або виявлення ефективних ознак, класів та правил їх розрізнення), в МГУА така задача називається «Об'єктивна Комп'ютерна Кластеризація» (ОКК);
- об'єктивний системний аналіз (ОСА), коли необхідно з'ясувати, які серед вимірюваних змінних є незалежними (вхідними), які залежними (вихідними), які зайвими (неінформативними), і побудувати відповідну модель тощо.

Індуктивне моделювання за статистичними даними є процесом послідовного прийняття рішень, що складається з певної сукупності послідовно

виконуваних етапів. Кожна з цих складових має свої параметри, характеристики та області допустимих значень, що дозволяє використовувати онтології для їх опису. В результаті такої структуризації визначаються [8] принципи формування алгоритмічних модулів для розв'язання конкретної задачі. В залежності від типу задач повинні бути вибрані адекватні методи їх розв'язання.

При конструюванні систем, заснованих на знаннях в різних галузях людської діяльності при розробці баз знань, як невід'ємної частини таких систем, часто застосовуються методи міркувань на основі прецедентів – *прецедентний підхід* (CBR – Case-Based Reasoning). Міркування на основі прецедентів базуються на використанні накопиченого досвіду та адаптації рішення відомої задачі для розв'язання нової. Використання онтологій для представлення прецеденту дозволяє задавати складну структуру, що може містити дані різних типів, забезпечити просте для розуміння представлення структурованих знань та спростити їх оновлення.

В загальному випадку модель представлення прецеденту містить опис ситуації та рішення для неї [9]:  $CASE = (Situation, Solution, Result)$ , де *Situation* – ситуація, що описує прецедент, *Solution* – рішення (наприклад, модель або прогноз та рекомендації для людини, що приймає рішення), *Result* – результат застосування рішення, який може містити перелік виконаних дій, коментарі, пояснення щодо вибору рішення та можливі альтернативи тощо.

В більшості випадків достатньо параметричного представлення прецеденту у вигляді набору параметрів з конкретними значеннями:

$CASE = \{x_1, \dots, x_n, R\}$ , де  $x_1, \dots, x_n$  - параметри конкретної ситуації ( $x_1 \in X_1, \dots, x_n \in X_n$ ),  $n$  – кількість параметрів,  $R$  – рішення (рекомендації ЛПР),  $X_1, \dots, X_n$  - області можливих значень відповідних параметрів, задані відповідними онтологіями.

Нехай

$KM$  – онтологія класів моделей,  $KM = \{k_i\}, i = \overline{1, K}$  ;

$GS$  – онтологія генераторів структур моделей,  $GS = \{g_j\}, j = \overline{1, G}$  ;

$OP$  – онтологія методів оцінювання параметрів структур,  
 $OP = \{p_q\}, q = \overline{1, Q}$  ;

$CR$  – онтологія критеріїв якості моделей,  $CR = \{r_t\}, t = \overline{1, R}$  .

$OR$  - множина онтологічних співвідношень.

Кожна з цих онтологій представляє собою множину імен (понять, термінів)

та їхніх характеристик (атрибути, параметри конкретної ситуації).

Наприклад, онтологія класів лінійних моделей  $KM$  визначає різні типи моделей з ключовими параметрами  $\{MY, LY, MX, LX\}$ , де  $MY, MX$  - кількість вхідних та вихідних змінних, а  $LY, LX$  - кількість врахованих минулих значень (запізнювань) для вхідних та вихідних змінних відповідно. В залежності від конкретних значень цих параметрів можна отримати більшість варіантів лінійних моделей, які зустрічаються на практиці, як одновимірних, так і багатовимірних [10].

Онтологія генераторів структур моделей  $GS$  містить два основних типи генераторів структур: перебірні (комбінаторні) та ітераційні (багаторядні) та характеристики ключових параметрів  $\{SI, SA, F, NI\}$ , що узагальнює низку типів генераторів структур (які відносяться не тільки до МГУА):

- багатовимірний регресійний аналіз ( $SI=SA=m$ );
- повний (комбінаторний) перебір (або алгоритм усіх регресій) ( $SI = 1, SA = m, F_s = C_m^s$ );
- регресійна процедура «включення» ( $F = 1$ );
- селекційно-комбінаторний генератор ( $1 < F < C_m^s$ );
- багаторядна ітераційна процедура МГУА ( $SI=SA=2, NI > 1$ ), тощо.

Ключові параметри  $\{\eta_1(\cdot), \eta_2(\cdot), V(s), \hat{\sigma}^2\}$  характеризують множину критеріїв, які застосовуються на практиці з задачах структурної ідентифікації моделей і визначаються заданням онтології критеріїв якості моделей  $CR$  (тут  $\eta_1(s,n), \eta_2(s,n)$  - мультиплікативний та адитивний члени, що мають зміст функцій штрафу за складність моделі,  $V(s)$  - деякий показник якості моделі,  $\hat{\sigma}$  - оцінка невідомої дисперсії  $\sigma^2$ ) [10].

Таким чином, множину методів побудови моделей можна задати у вигляді  $\langle KM, GS, OP, CR, OR \rangle$ . Тобто, вони можуть задаватись певним набором значень відповідних онтологій. Різні комбінації цих параметрів утворюють окремі методи як окремі випадки, прецеденти. При такому підході формування компонент інтерфейсу користувача зводиться до задання конкретних значень понять відповідних універсальних моделей онтологій.

### Висновки.

В роботі наведені принципи побудови інтерфейсу користувача на основі моделей онтологій при розробці комп'ютерних систем, заснованих на знаннях. Такий підхід дає можливість спростити конструювання ефективних, дружніх до користувачів різного рівня підготовки інтерфейсів, які мають ознаки інтелектуального інтерфейсу. Високий рівень формалізму онтологій, їхня незалежність від мов та засобів програмування, можливість внесення змін в структуру інтерфейсу незалежно від інших частин прикладної програми,



автоматичне кодування на основі онтологій надає переваг такому підходу.

### Література:

1. Хорошевский В.Ф. Семантические технологии: ожидание и тренды. // Сборник докладов OSTIS-2012, Минск: БГУИР, 2012. ISBN 978-985-488-683-1 – С. 141-158. (<http://conf.ostis.net>)
2. Поспелов Д.А. Интеллектуальные интерфейсы для ЭВМ новых поколений. // Электронная вычислительная техника. Сборник статей. Вып.3, М.: Радио и связь, 1989, с.4-20.
3. Степашко В.С., Зворигіна Т.Ф., Піднебесна Г.А. Знання-орієнтований підхід до конструювання інтерактивних засобів індуктивного моделювання. // Обчислювальний інтелект (результати, проблеми, перспективи): Матеріали 1-ї Міжнародної науково-технічної конференції (10-13 травня 2011, м. Черкаси). – Черкаси, Маклаут, 2011. – С. 246.
4. Степашко В.С. О задаче структуризации знаний эксперта в области моделирования по эмпирическим данным // Кибернетика и вычислительная техника. - 1991.- Вып. 92.- С.80-83.
5. Гаврилова Т.А. Онтологический подход к управлению знаниями при разработке корпоративных систем автоматизации. // [http://bigc.ru/publications/bigspb/km/ontol\\_podhod\\_to\\_uz.php](http://bigc.ru/publications/bigspb/km/ontol_podhod_to_uz.php).
6. Грибова В.В, Клещеев А.С. Онтологическая парадигма программирования. // конференція OSTIS-2012 (Open Semantic Technologies for Intelligent Systems), 2012. – С. 213-220. (<http://conf.ostis.net>).
7. Піднебесна Г.А. Концепція використання онтологій для конструювання засобів індуктивного моделювання . // Індуктивне моделювання складних систем. Збірник наук. праць. К.:МННЦІТС, 2013, вип. 5. С. 248-255.
8. Степашко В.С., Савченко Є.А., Піднебесна Г.А. Індуктивне моделювання як процес послідовного прийняття рішень // Матеріали міжнародної наукової конференції «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту» (ISDMCI'2010), Євпаторія. – Херсон: Видавництво ХНТУ, 2010. – Т.2. – С. 457-460.
9. Варшавский П., Алехин Р. Метод поиска решений в интеллектуальных системах поддержки принятия решений на основе прецедентов. // International Journal "Information Models and Analyses" Vol.2 / 2013, Number 4. - С. 385 – 392. (<http://www.foibg.com/ijima/vol02/ijima02-04-p09.pdf>).
10. Степашко В.С., Піднебесна Г.А. Концепція узагальнених багатофункціональних модулів як основа конструювання засобів індуктивного моделювання// Індуктивне моделювання складних систем. Збірник наук. праць. К.:МННЦІТС, 2011, вип. 3. - С. 215-222.