

5. Олецкий О. В. Побудова багатокomпонентної онтологічної системи для автоматизованого експертного добору навчальних матеріалів // Теоретичні та прикладні аспекти побудови програмних систем. Матеріали міжнародної конференції ТАAPSD'2009, 8–10 грудня 2009 р. – С. 83–88.
6. Ландэ Д. В. Поиск знаний в Internet / Д. В. Ландэ. – М. : Вильямс, 2005. – 272 с. – ISBN 5-8459-0764-0.
7. Сэлтон Г. Автоматическая обработка, хранение и поиск информации / Г. Сэлтон. – М. : Сов. радио, 1973. – 560 с.
8. Плєскач В. Л. Агентні технології / Плєскач В. Л., Рогушина Ю. В. – К. : Київ. нац. торг.-екон. ун-т, 2005. – 338 с.
9. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткая логика / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. : Горячая линия – Телеком, 2004. – 452 с. – ISBN 5-93517-103-1.
10. Зайченко Ю. П. Основи проектування інтелектуальних систем / Ю. П. Зайченко. – К. : Слово, 2004. – 352 с.
11. Николенко С. И. Самообучающиеся системы / С. И. Николенко, А. Л. Тулупьев. – М. : МЦНМО, 2009. – 288 с. – ISBN 978-5-94057-506-1.
12. Глибовец Н. Н. Генетические алгоритмы и их использование для решения задач составления расписания / Глибовец Н. Н., Медведь С. А. // Кибернетика и системный анализ. – 2003. – №1. – С. 95–108.
13. Глибовець М. М. Про деякі підходи до проблеми інформаційного керування випадковим пошуком / Глибовець М. М., Олецкий О. В. // Dynamical System Modelling and Stability Investigation. Thesis of Conference Reports, May 22–25, 2007. – С. 370.

*O. Oletsky*

## PRINCIPLES OF USING GENETIC ALGORITHMS FOR ONTOLOGY ORIENTED INFORMATION SEARCH

*Recommendations for using genetic algorithms for ontology-based information search are formulated. Parameters of the ontology-based model of content are classified; relations, parameters of which should be optimized, are proposed. Some approaches for choosing optimization criterion are regarded.*

**Keywords:** Genetic algorithm, information search, ontology analysis.

УДК 004.415.2

*Парфірова Т. С.*

## ПОНЯТТЯ ВЗАЄМОДІЇ З ТОЧКИ ЗОРУ СУТНІСНОЇ ПЛАТФОРМИ

*Проведено розгортання поняття взаємодії із застосуванням методології сутнісної платформи. Отримано модель взаємодії, достатньо загальну для охоплення конкретних типів взаємодій. Загальність моделі підтримується тим, що вона містить механізми для саморозгортання і конкретизації.*

**Ключові слова:** сутнісна платформа, взаємодія, активна сутність, пасивна сутність, результат взаємодії, прикладні засоби.

У роботах В. Н. Редько та його наукової школи [1–3] сформульовано основи сутнісної платформи (СП), які стали продовженням попередніх робіт, направлених на побудову концептуально єдиної теорії програмування.

Нагадаємо основні поняття СП. Сутність – все, що існує, в найзагальнішому сенсі цього слова. Деякі сутності виступають сутями інших сутностей, тобто є їхніми прагматико-обумовленими властивостями. Під властивістю розуміють сутність, яка характеризує сутність. Поняття

суттєвості відношення відіграє виключно важливу роль у побудові сутнісної платформи.

СП носить характер загальної теорії, яка, з одного боку, описує все, а з другого, її повнота носить не актуальний, а потенційний характер. Тобто яку б область дійсності не взяти, хоча СП не дає її теоретичної моделі в готовому вигляді, проте можна провести розгортання СП до адекватної теорії, яка описує цю область. Будь-яке розгортання в СП повинно відповідати принципу достатніх основ, введення абстракції може

здійснюватися лише у зв'язку з необхідністю, обумовленою прагматикою.

Цю статтю присвячено розгляду одного з важливих понять, що претендують на високий ступінь загальності. Серед різних відношень між сутностями особливу роль відіграють відношення взаємодії. Наприклад, такі відношення, як «частина–ціле», «рід–вид», «подібність» та багато інших – це, умовно кажучи, просторові відношення, вони характеризують стан дійсності у певний момент часу, а взаємодія додає аспект часу, динаміки, зміни стану. Сутності існують не самі по собі, їхнє існування, а особливо суть, розкриваються через їхні відношення, і насамперед (особливо в програмології, яка власне займається процесами) – через взаємодії. Завданням нашої роботи є поетапний розгляд поняття *взаємодії*, керуючись принципом достатніх основ.

Слід зазначити, що в такому формулюванні ця проблема у відомих роботах інших авторів не розглядалася. У теорії та практиці програмування досліджувалися різноманітні часткові випадки та конкретні різновиди взаємодії: наприклад, з теоретичних здобутків згадаємо модель взаємодії послідовних процесів CSP – Communicating sequential processes [4], а з практичних технологій – програмування, кероване подіями, Event-driven programming [5-8]. Але, з другого боку, детальний розгляд теорії чи техніки реалізації окремих типів взаємодій в цих роботах не супроводжується підйомом рівня абстракції до розгляду власне взаємодії, взаємодії як такої, в загальному випадку. Це, своєю чергою, не дозволяє поставити питання про співставлення, порівняння чи інтеграцію тих чи інших окремих типів взаємодій, бо для цього потрібно звести ці різні типи до спільної концептуальної бази; крім того, це не дає можливості з єдиної концептуальної основи цілеспрямовано знаходити нові конкретні механізми взаємодій.

У першому наближенні взаємодія експлікується як дія, в яку втягнуті декілька сутностей (без втрати загальності можна обмежитись двома сутностями).

Властивість сутностей «взаємодіяти» помітно багатша за властивість «діяти». Дійсно, можна розглядати односторонні дії (це або дії сутності на саму себе, або дії сутності в певному середовищі, яке відповідно до прагматики виводиться з розгляду). Розгляд же *спільної (сумісної)* дії потребує врахування наявності спільного інтерфейсу двох сутностей – таких їхніх властивостей та відношень, які роблять можливою участь у єдиному процесі.

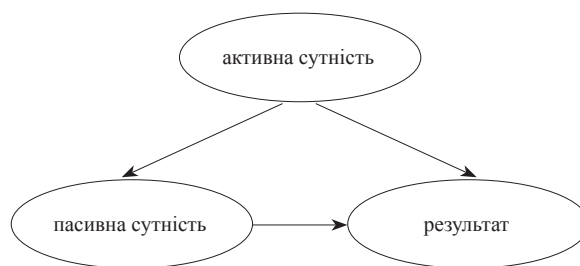
Також відмітимо, що навіть говорячи про взаємодію двох сутностей, мається на увазі і третя – *результат* взаємодії. У якості результату може виступати не лише об'єкт, але і процес,

зміна стану однієї чи двох взаємодіючих сутностей і т. д. – відповідно до сутнісної платформи, все це також можна вважати сутностями.

Для подальшого розгляду варто провести ще одну обмовлену прагматикою конкретизацію. Будемо вважати загальнозначущою властивістю те, що сутності відіграють у взаємодії помітно різні ролі. Без такої конкретизації ми були б обмежені розглядом тривіальних однорангових взаємодій і були б позбавлені можливості адекватно моделювати ті види взаємодії, які відіграють найбільш важливу роль у практиці.

На перший погляд, будь-які дві сутності пов'язані між собою апріорі незліченною кількістю способів і вступають у взаємодію багатьох різних видів, відіграючи свою роль у кожній із взаємодій. Усе різноманіття ролей, безумовно, не варто виносити на загальнозначущий рівень: конкретні ролі носять предметний характер і специфічні для свого класу задач. Важливо, що всі предметно-специфічні пари ролей можуть бути експліцитно зведені до однієї загальнозначущої пари «активний – пасивний».

Отримана на певному кроці експлікації модель «активна сутність – пасивна сутність – результат» (мал. 1) все ще носить статичний характер і не відображає *процес* дії. Хоча дія в цій моделі і згадується, але не розкривається її природа, структура, спосіб протікання, тобто вона присутня у згорнутому, інкапсульованому вигляді. Тому необхідно доповнити модель застосування однієї сутності до іншої.



Мал. 1. Модель «активна сутність – пасивна сутність – результат»

Відомо багато моделей, які в подробицях і деталях описують процеси застосування обмежених класів активних сутностей: Машина Тьюрінга, автомати з магазинною пам'яттю, операціональні семантики мов програмування і т. д. Проте кожна з цих моделей орієнтована на порівняно вузьку область застосування і характеризується надто високим рівнем деталізації, що робить неможливим їхню загальнозначимість у сенсі СП.

Лейбніцевому принципу достатніх основ відповідає лише мінімально необхідне пониження абстрактності: загальнозначущим є факт існу-

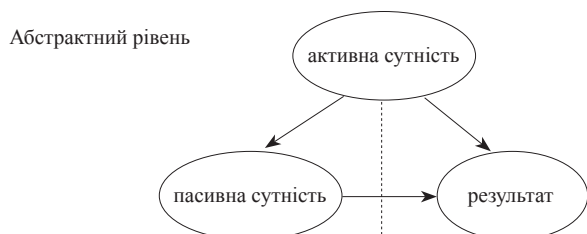
вання певного *способу застосування* активної сутності до пасивної, тоді як природа і логіко-математичне визначення цього способу специфічні для кожного конкретного випадку і витісняються на предметний рівень. Таким чином, вказана вище тричленна модель може бути зведена до чотирьохчленної: «активна сутність – пасивна сутність – спосіб застосування – результат» (мал. 2).



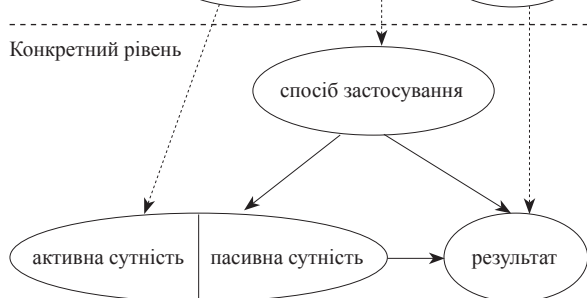
**Мал. 2.** Модель «активна сутність – пасивна сутність – спосіб застосування – результат»

Зазначимо, що біполь «активна сутність – пасивна сутність», відповідно до гранично загального тлумачення сутностей в СП, можна також розглядати як сутність; сутністю є також і спосіб застосування. Це означає, що модель, наведена на мал. 2, є конкретизацією моделі з мал. 1, де в якості абстрактної активної сутності взято спосіб застосування, а в ролі абстрактної пасивної – біполь конкретної активної конкретної і пасивної сутностей (мал. 3).

Абстрактний рівень



Конкретний рівень



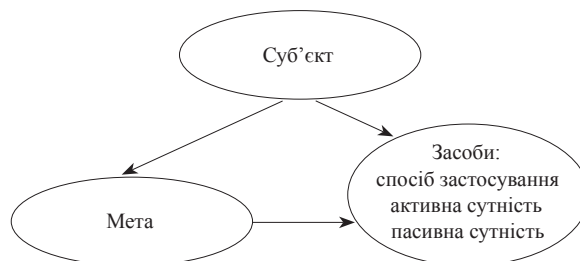
**Мал. 3.** Саморозгортання трьохчленної моделі до чотирьохчленної

Подальший виток розгортання сутесутнісної моделі взаємодій пов'язаний з введенням дуже сильної додаткової конкретики, яка, однак, цілком виправдана прагматикою. Наведена модель відображає взаємодію сутностей з точки зору са-

мих лише сутностей – учасників взаємодії, та з точки зору того моменту часу, коли взаємодія вже відбулася і її можна описувати ретроспективно. Але ж для того, щоб активна сутність подіяла на пасивну певним чином, потрібно спершу якось здійснити вибір і активної, і пасивної сутності, і способу застосування з-поміж невичерпного різноманіття усіх сутностей. Загалом можна уявити безліч механізмів такого вибору. Зокрема, вибір може здійснюватися спонтанно, як у стохастичних процесах, або як у механічній системі, жорстко визначатися її будовою. Проте в контексті теорії програмування як науки про цілеспрямовані процеси створення програм нас цікавить усвідомлений, цілеспрямований вибір сутностей певним суб'єктом для певної мети.

Важливо, що хоча процес вибору суб'єктом засобів для досягнення мети може розглядатися з різним ступенем деталізації та з різних точок зору, проте всі отримані при цьому моделі виявляються конкретизаціями тричленної моделі з мал. 1.

На низькому рівні деталізації маємо, що суб'єкт (активна, творча сутність), виходячи з певної мети (яка хоча й спрямовує зусилля суб'єкта, сама не діє і тому розглядається як пасивна сутність), отримує засоби для досягнення мети – триєдину сутність, що включає в себе активну, пасивну сутність та спосіб застосування другої до першої (мал. 4).

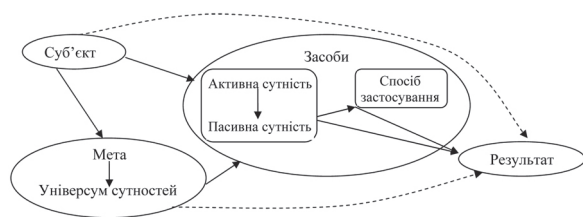


**Мал. 4.** Модель «Суб'єкт – мета – засоби»

З дещо більшою деталізацією можна сказати, що вибір засобів здійснюється з певного універсуму сутностей, який відіграє пасивну роль щодо мети і суб'єкта, який шукає в ньому певні корисні для заданої мети сутності (мал. 5).



**Мал. 5.** Деталізована модель «Суб'єкт – мета – засоби»



Мал. 6. Розгорнута модель досягнення мети суб'єктом

Нарешті, об'єднання моделі взаємодії суб'єкта з метою і універсумом у пошуках засобів та моделі взаємодії знайдених засобів дає розгорнуту модель досягнення суб'єктом мети (мал. 6). Спочатку суб'єкт, який має мету і знаходиться у фіксованих межах (універсумі сутностей), знаходить засоби (досягає проміжної мети), а потім, застосувавши знайдений активний засіб до пасивного знайденого способом, отримує остаточний результат, який і відповідає початковій меті.

Таким чином, тричленна модель «активна сутність – пасивна сутність – результат» в рам-

ках сутнісної платформи виявляється достатньо загальною, щоб описати широкий клас явищ принципово різного масштабу та ступеню деталізації залежно від того, що береться в якості активної та пасивної сутності. Крім того, ця модель здатна до розгортання; механізмом такого розгортання є подрібнення сутностей, коли те, що на попередньому рівні розгляду поставало як одна сутність, починає розглядатися як бі-поль, в якому можна виділити свою активну і пасивну частини. Це дає підстави назвати наведену схему еталонною моделлю взаємодій. Таким чином, спільну концептуально-методологічну основу для аналізу та синтезу різноманітних конкретних типів взаємодій, про яку говорилося на початку роботи, побудовано. Подальший розвиток дослідження передбачає застосування еталонної моделі до репрезентативних прикладів взаємодій, що використовуються на практиці при побудові програмних систем, в першу чергу – до чисто послідовної взаємодії.

1. Редько В. Н. Экзистенциальные основания композиционной парадигмы / В. Н. Редько // Кибернетика и системный анализ. – 2008. – № 2. – С. 3–12.
2. Редько В. Н. Программологические основания сущностной платформы / Редько В. Н., Редько И. В., Гришко Н. В. // Проблемы программирования. – 2008. – № 2–3 (Спец. выпуск). – С. 75–83.
3. Редько І. В. Теорія дескриптивних середовищ та її застосування: автореф. дис. на здобуття наук. ступеня доктора фіз.-мат. Наук : спец. 01.05.03 «Математичне та програмне забезпечення обчислювальних машин і систем» / Редько Ігор Володимирович. – К., 2008. – 32 с.
4. Хоар Ч. Взаимодействующие последовательные процессы / Хоар Ч.; пер. с англ. – М. : Мир, 1989. – 264 с.: ил. – ISBN 5-03-001043-2.
5. Ануфриев О. В. О методике обучения основам событийного программирования [Электронный ресурс]: Материалы 8-й

- Международной научно-методической конференции «Новые информационные технологии в университетском образовании» / О. В. Ануфриев // ИКТ портал. – Режим доступа : <http://www.ict.edu.ru/vconf/files/3195.rtf>.
6. Непейвода Н. Н. Стили и методы программирования. Курс лекций. Учебное пособие / Н. Н. Непейвода. – М. : Интернет-университет информационных технологий, 2005. – 316 с.
7. Ferg St. Event-Driven Programming: Introduction, Tutorial, History [Электронный ресурс] / Ferg Stephen. – Version 0.2 – 2006-02-08 // Режим доступа : [http://eventdrivenpgm.sourceforge.net/event\\_driven\\_programming.pdf](http://eventdrivenpgm.sourceforge.net/event_driven_programming.pdf)
8. Fowler M. Event Collaboration [Электронный ресурс] / Martin Fowler // Режим доступа: <http://www.martinfowler.com/eaDev/EventCollaboration.html>.

T. Parfirova

## INTERACTION NOTION IN TERMS OF ENTITY PLATFORM

*Explication of the interaction notion is carried out on the basis of entity platform. An interaction model is developed, general enough to encompass various concrete kinds of interaction. Generality of this model is supported by the fact that it contains inherent means of self-deployment and self-concretisation.*

**Keywords:** entity platform, interaction, active entity, passive entity, interaction result, application means.