

СУТЕСУТНІСТНІ ХАРАКТЕРИСТИКИ НАВЧАЛЬНИХ СЕРЕДОВИЩ

Т.С. Парфірова

Київський національний університет імені Тараса Шевченка,
Київ, 01601, вул. Владимирська, 64, eremita.t@gmail.com

Проведено аналіз репрезентативних прикладів навчальних середовищ з використанням методології сутнісної платформи. Виявлено, що вони не повністю адекватно підтримують потреби реального навчального процесу. Виділено основні характеристичні властивості навчальних середовищ. Запропоновано інтеграційний підхід до розробки навчальних середовищ як відкрито-замкнених систем.

Representative examples of educational environments are analyzed on the methodological basis of entity platform. It is revealed they do not completely and adequately support needs of real educational process. Basic characteristic features of educational environments are discovered. Integration approach to development of educational environments as open-close systems is proposed.

Методологічні засади дослідження сутністних характеристик навчальних середовищ

У роботі [1] показано, що структура та побудова програмного комплексу (в тому числі й навчальних середовищ) мають бути такими, щоб у них чітко виділялись логічний інваріант (ядро) і те, що носить змінний, допоміжний характер (у роботах [2, 3] – логічне і предметне). Ядро є замкненим і забезпечує орієнтацію програмної системи на певний клас задач. Варіативна (предметна) частина втілює відкритість системи, її здатність до гнучкого налаштування на будь-яку задачу з цього класу. Типовим прикладом інваріантного та змінного в сучасних системах можуть бути вихідний код та конфігураційні файли: зміна вихідного коду веде до появи нової системи, а параметри, задані у конфігураційних файлах, можна змінювати гнучко під час її роботи. Іншим прикладом є ядро операційної системи (зміна лише шляхом перекомпіляції) та модулі, які можна відключати та підключати під час роботи.

Звернемо увагу на зв'язок логічного та предметного. Саме у замкненій логічно частині потенційно закладена можливість варіативної частини. Розподіл на логічне і предметне є відносним і обумовлюється прагматикою. Наприклад, для розробників Linux-подібних ОС замкнена частина – ядро (kernel), а набір прикладних програм – предметна частина, для конкретного користувача все це складає „замкнену частину”, а варіативність проявляється в конкретних налаштуваннях типу вид меню, вікон і т. д.

Також у роботі [1] показано, що програмні комплекси (в тому числі й навчальні середовища) мають у собі органічно поєднувати як відкритість, так і замкненість, тобто бути відкрито-замкненими). Загальний метод аналізу та синтезу відкрито-замкнених систем розроблено в працях В.Н. Редька та І.В. Редька. Нагадаємо основні його положення. За основу розгляду береться гранично загальне поняття сутності, як будь-чого, що в деякому розумному сенсі існує. Ключове місце в теорії дескриптивних середовищ займає індивідуалізація прагматико-обумовлених типів в універсумі сутностей на основі *сутесутнісних відношень* – відношень між прагматико-обумовленими властивостями сутностей (сутями) і сутностями, що володіють цими властивостями (доменами) [2]. Прагматика визначається відношенням підпорядкованості. В основі розділення на логічне і предметне лежать *сутесутнісні відношення*. Засіб їх використання – метод сутесутнісної релятивізації (MCP) [2]. При розгляді акцент робиться не на самих сутностях, а на індукованих ними прагматико-обумовлених логіко-предметних відношеннях. Речі розглядаються як вузли в сітці відношень. Серед усіх сутностей особливо важливе місце займають сутності у вигляді яких існують інші сутності, тобто екзистонні суті.

Для побудови дескриптологічної моделі предметної галузі на першому кроці треба виділити множину сутностей та з'ясувати, як вони впорядковані між собою сутесутнісними відношеннями. Сутесутнісне відношення є універсальній засіб синтезу сутностей.

Основа MCP складають універсальні в класі суб'єктів (інтерсуб'єктивні, тобто незалежні від того, хто їх використовуватиме) співвіднесення сутей з прагматикою, які визначають сутесутнісну релятивізацію сутностей із універсуму сутностей (УС), що є несуперечливою абстракцією цілісного різноманіття різних сутностей [3]. У логічне ядро поміщають домени (сутності, які володіють сутями), суті складають предметну частину.

Сутності предметної галузі та їх підтримка в наявних середовищах

Виділимо, згідно вищеописаної методології, характерні для останньої галузі сутності разом з їх взаємозв'язками. Ці сутності мають покривати прагматико обумовлені аспекти, не обтяжені конкретикою, у тому розумінні, що не включаємо надто дрібні сутності, які не носять системного характеру, а з іншого боку не бути обтяженими абстрактністю (сутності типу „свідомість”, „буття”).

Системний аналіз реальних освітніх процесів дозволяє сформувати в першому наближенні перелік основних сутностей, наведених у таблиці, де показано як ці сутності підтримуються в репрезентативних

© Т.С. Парфірова, 2010

програмних продуктах.

З таблиці видно, що типові представники даного класу програмних систем при всіх їх відмінностях або однаково підтримують, або однаково не підтримують ті чи інші сутності (хоча характер, ступінь і спосіб підтримки можуть суттєво відрізнятися). Це означає, що на певному, доволі високому рівні абстракції, можна вважати, що ці програмні продукти втілюють одну концептуальну модель освітньої галузі.

Таблиця

Сутність \ НС	Moodle	Піас	Learning Space 5.x	Lotus Learning Management System (LMS)	IBM Workplace Collaboration
Дисципліна \ назва курсу	+	+	+	+	+
Іспит	+	+	+	+	+
Студент	+	+	+	+	+
Група	+	+	+	+	+
Викладач	+	+	+	+	+
Вид заняття	+	+	+	+	+
Форма проведення	+	+	+	+	+
Вид матеріалу	+	+	+	+	+
Форма матеріалу	+	+	+	+	+
Критерії оцінювання	+	+	+	+	+
Форми контролю	+	+	+	+	+
Залежність дисциплін	-	-	-	-	-
Навчальний план	-	-	-	-	-
Керуючий орган	-	-	-	-	-
Ієрархічні зв'язки між викладачем і керуючим органом	-	-	-	-	-

Розглянемо ті сутності, підтримка яких відсутня в наявних навчальних середовищах (НС). Залежність дисциплін важлива, коли навчальний план будується з кількох різних курсів, у той час як у процесі роботи з конкретним курсом поняття залежності інкапсульоване (тобто не виступає самостійним предметом обробки). Управлінським органом може виступати завідувач кафедри, деканат, ректорат і відігравати важливу роль лише на макрорівні, здійснюючи узгодження діяльності окремих викладачів, розподіл ресурсів між потоками студентів, контроль якості роботи викладачів тощо. Водночас на мікрорівні процес проведення кожного конкретного заняття не потребує постійної уваги управлінців. Розглянуті середовища підтримують розклад занять на рівні конкретного курсу та не дозволяють складати комплексний розклад.

Таким чином можемо сформулювати висновок, що дані навчальні середовища тією чи іншою мірою підтримують викладання на рівні окремого курсу, але безпосередньо не містять засобів для організації робіт у масштабі закладу. Це могло б бути допустимим, якби НС містили природні й гнучкі засоби розширення, що дозволяли б за потребою інтегрувати їх з іншими програмними системами, таким чином поповнюючи їх концептуальну модель та функціональні можливості. Розгляду даного питання присвячено наступний розділ.

Інтеграційний погляд на навчальні середовища

Варто розрізнити НС у вузькому (система, що підтримує власне навчальний процес) та в широкому розумінні (комплекс, що крім навчального процесу підтримує й інші, пов'язані з ним аспекти діяльності закладу: кадровий, бухгалтерський облік, керування господарством тощо). НС у вузькому розумінні являють собою навчальні системи, в широкому сенсі – це системи управління навчанням.

У контексті даної роботи нас цікавлять НС у вузькому сенсі (системи автоматизації навчання), але такі, які б підтримували інтеграцію до рівня систем управління навчанням (типу IBM Lotus LMS) або ж до систем автоматизації управління підприємством (типу IBM WorkPlace), назвемо її інтеграцією „вверх” (рисунок). Крім цього має бути підтримка інтерфейсів „вниз” – можливість внесення змін у саму навчаючу складову без зміни основної функціональності, а лише шляхом перебудови, підключення, модифікації допоміжних компонент. Якщо перші два варіанти цілком очевидні, то увага третій приділяється значно менша. Крім інтерфейсів до вищого рівня (керуючого) й інтерфейсів до нижчого (допоміжного) важливими є внутрішні інтерфейси, призначені для організації взаємодії в межах поточного структурного рівня.

Важливість цих інтерфейсів обґрунтовується наступними міркуваннями. Можливості технологій проектування та програмування відстають від складності розроблюваних систем, а це призводить до того, що на етапі випробувань виявляються невідповідності програмних систем щодо предметної області та сформульованих попередньо вимог, значить і до необхідності внесення змін.

Якщо логікою побудови системи не передбачена можливість внесення локалізованих коректив в окремі слабозв'язані компоненти, то навіть дрібна модифікація системи може викликати каскад змін у її підсистемах, зачепити інтерфейси між ними й спотворити інтерфейси «вниз» і «вгору». Таким чином інтерфейси середнього рівня утворюють логіку системи, є її інваріантом. Варто зазначити, що саме стійкі взаємозв'язки, а не компоненти утворюють структуру системи.

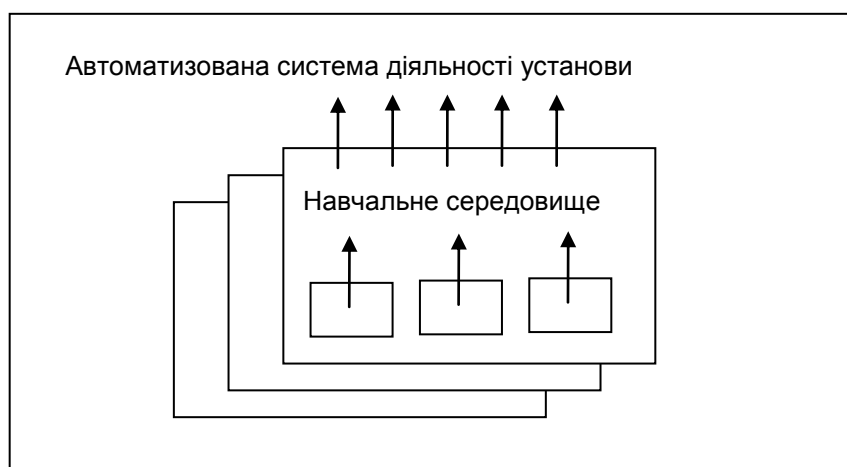


Рисунок. Навчальне середовище як частина інтеграційного

Спробуємо критично оглянути реальний стан справ. Moodle і Pias, наприклад, забезпечують розробку курсів, контроль знань, інтерактивність спілкування Викладач-Студент, але не можуть бути органічно включені в систему управління навчальним процесом загалом, а тим більше в автоматизовану систему управління деяким підприємством, основним видом діяльності якого не обов'язково є навчальна. Звичайно, кожна з названих систем відповідно до нових стандартів і вимог зобов'язана надавати таку можливість інтеграції, але реалізація цього здійснюється, як зазначено в [4–11], не на логічному рівні побудови самої системи, а на рівні API. Тому в кожному конкретному випадку навіть вбудоване НС виглядає як „інородне тіло”, що веде до проблем адекватної підтримки навчального процесу в цілому.

Для подолання або принаймні зменшення розриву між розроблюваними навчальними середовищами і реальними потребами процесу, який вони мають підтримувати, пропонується реалізація навчальних середовищ відкрито-замкнених систем.

Основні характеристичні властивості навчальних середовищ

При дослідженні репрезентативних прикладів НС і ступеню адекватної підтримки ними предметної галузі виокремлено наступні властивості.

Високий ступінь формалізованості ділових стосунків, характерний для предметної області. Тобто в реальній діяльності, підтримку якої забезпечують ці середовища, нормативні документи регламентують поведінку кожного суб'єкта та системи в цілому. Також чітко регламентовані форми звітності й контролю.

Ще однією характеристикою є неперервно-поточний процес. Середовище має підтримувати не розв'язання деякої обмеженої задачі (такі програми припиняють роботу як тільки задача розв'язана і отримано результат), а розгорнуту в часі діяльність, яка не має наперед визначеної точки повного закінчення.

Такі середовища забезпечують автоматизацію наявного реального „бізнес-процесу”, а не моделюють можливі наслідки чи результати перебудов, реструктуризацій.

Окрім вищеперерахованих, у середовищ є й інші визначальні характеристики. Зокрема, складні внутрішні взаємозв'язки (механізми передачі даних, обміну інформацією, результатами роботи підсистем, обробки даних і т. д.), тоді як зовнішній клієнт простий з точки зору парадигми розробки.

На відміну від різного роду обчислювальних програм або систем, які, наприклад, управляють складними технологічними процесами, для розглянутих середовищ характерним є незначний відсоток складних обчислень, але велика кількість запитів до БД.

Також НС мають забезпечувати роботу в реальному режимі часу (реальним режимом часу вважаємо швидкість роботи системи, яка збігається зі швидкістю протікання процесів у об'єкті управління).

Наступну групу характеристик можна об'єднати властивістю дискретності, а саме – для середовищ характерна дискретність керованого об'єкта за часом. Процес складається з окремих подій (горизонтальна дискретність). Дискретність за станом означає, що фазовий простір містить скінчену множину станів (вертикальна дискретність).

Зазвичай у таких середовищах чітко прослідковується дискретність за статикою: наявність ієрархічної структури. При цьому наявні як висхідні, так і низхідні інформаційні зв'язки, що забезпечує зворотній зв'язок і, як результат, адаптивність системи. Керуючі зв'язки є низхідними, що власне і формує чітку ієрархію підпорядкованості.

Дискретність за процесом проявляється в розбитті завдання на окремі підзадачі та розгортанні його для підсистем (наприклад, завдання покращення успішності для кожного підрозділу буде конкретизовано по-іншому).

Обов'язковою є підтримка обліку поточних характеристик учасників процесу з періодичним оновленням інформації й подальшими діями з управління, які виконуються у відповідності зі станом кожного учасника процесу.

Враховуючи дані властивості можна зробити висновок. Суть НС, з одного боку, визначається специфікою галузі освіти та навчання, з другого – НС є репрезентативним прикладом для більш широкого класу систем, який назвемо середовищами підтримки організаційної діяльності. Звичайно НС не зводяться повністю до таких середовищ, проте значна частина функціональності є спільною. Сутнісна подібність різних організаційних задач зумовлює близькість функціональних вимог до програмних систем, призначених для їх автоматизації, а це – спільність технологій і архітектурних рішень. Таким чином, єдність означеного класу підтримується на трьох рівнях: прагматика, поведінка та внутрішня побудова. Тому розгляд суттєвих відношень та визначальних властивостей НС може бути частково перенесеним і використаним для класу середовищ підтримки організаційної діяльності.

Висновки

Використовуючи методологічний апарат, запропонований у [2–4], проведено аналіз репрезентативних прикладів навчальних середовищ. Виділено найбільш характерні сутності предметної галузі й виявлено як вони підтримуються розглянутими навчальними середовищами. На основі цього аналізу обґрунтовано, що навчальне середовище повинне мати засоби інтеграції до більшої системи автоматизації діяльності закладу, причому наявність цих інтерфейсів має закладатись у логіку середовища на етапі проектування. Цього бракує розповсюдженим програмним продуктам з даної галузі. Для усунення цього недоліку доцільно розробляти такі навчальні середовища як відкрито-замкнені системи.

1. *Парфірова Т.С.* Навчальні середовища як відкрито-замкнені системи // Вісн. Київськ. ун-ту. Серія: фіз.-мат. науки. – 2009. – № 1. – С. 127–132.
2. *Редько І.В.* Теорія дескриптивних середовищ та її застосування: Автореф. дис. ... д-ра фіз.-мат. наук. – К., 2008. – 32 с.
3. *Редько І.В.* Дескриптологическая среда моделирования предметных областей // Проблемы програмування. – 2002. – №1–2 (Спец. випуск). – С. 44–57.
4. *Редько В.Н.* Основания программологии // Кибернетика и системный анализ. – 2000. – № 1. – С. 35–57.
5. *Declan Dagger et al.* Service-Oriented E-Learning Platforms. From Monolithic Systems to Flexible Services. IEEE Internet Computing, May–June 2007. – IEEE Computer Society, 2007.
6. *Moodle org: open-source community-based tools for learning* [Електронний ресурс]. – Режим доступу: <http://www.moodle.org>
7. *Ilias Open Source LMS* [Електронний ресурс]. – Режим доступу: <http://www.ilias.de>
8. *IBM collaboration software – Lotus software* [Електронний ресурс]. – Режим доступу: <http://www.lotus.com/learningspace>
9. *IBM Lotus Learning Management System* [Електронний ресурс]. – Режим доступу: <http://www-01.ibm.com/software/lotus/products/learning-management-system/>
10. *IBM Workplace Collaboration Services Home Page* [Електронний ресурс]. – Режим доступу: <http://www-01.ibm.com/software/lotus/products/wp-collaboration-services/>
11. *Tara Hall Lessons learned: Fernando Salazar and the LMS architecture 03 Feb 2003* [Електронний ресурс]. – Режим доступу: <http://www.ibm.com/developerworks/lotus/>