

ПОБУДОВА ПРОТОКОЛУ КОМУНІКАЦІЇ РОЗПОДІЛЕНИХ НАВЧАЛЬНИХ РЕПОЗИТОРІЇВ НА ОСНОВІ ECL

Визначення протоколу комунікації у розподіленому середовищі навчальних репозиторіїв є першочерговим завданням при побудові необхідної інфраструктури для ефективного обміну навчальними об'єктами. У статті відображено основні вимоги до протоколу, окреслено задачі, що має вирішувати протокол, а також запропоновано конфігурацію на основі eduSource Communication Language (ECL).

З подальшим розвитком динамічності веб-сторінок (перехід до Web 2.0 технологій [0]) та Інтернет-змісту в цілому дедалі більше навчальних матеріалів потрапляють у поле зору пересічного студента чи викладача. Попри велику кількість навчальних систем (НС), що використовують здебільшого власні формати для збереження модулів, надзвичайної популярності набув стандарт SCORM [2], що значною мірою полегшив збереження метаданих та створення навчальних об'єктів (НО) повторного використання. Проте головною проблемою сьогодення з навчальними матеріалами є, власне, їх якість, та застосовуваність до потреб учня, тобто врахування аспекту персоналізації. Зокрема, постають питання ефективного пошуку та використання повторюваних НО, відсіювання нерелевантних матеріалів, знаходження потрібних НО для конкретного курсу. Оскільки більшість навчальних сховищ не з'єднані потужним пошуковим двигуном на кшталт Google, нашим завданням є побудова такої інфраструктури комунікаційних протоколів, за допомогою яких можна об'єднати наявні репозиторії для ефективного знаходження та доставки релевантних НО.

Організації, що володіють одним або кількома НС чи репозиторіями НО, є, як правило, автономними. Цей принцип означає, що навчальний заклад може надавати доступ до власних ресурсів лише за умови того, що має над ними контроль. Надалі при визначенні архітектури інтероперабельного репозиторію та його інфраструктури ми враховуватимемо такі аспекти автономії окремих НС чи сховищ НО [3]:

1. Автономія **дизайну** - здатність окремої системи централізовано визначати: формат та вміст даних, їх концептуалізацію чи семантичну інтерпретацію, обмеження на використання даних, функціональність системи та інструментарій для її забезпечення (структура бази даних, алгоритми конкурентного доступу тощо).

2. Автономія **комунікацій** означає, що кожна система сама вирішує, чи спілкуватися їй з іншими компонентами певної мережі, а також використовує для цього визначений формат відповідей на запити.

3. Автономія **виконання** дозволяє власне системі визначати порядок виконання внутрішніх операцій, зважаючи на взаємодію із зовнішніми системами.

Специфікація IMS DRI [4], яка є найбільш повним документом, що описує архітектуру розподілених навчальних сховищ, як відомо, не дає конкретних вказівок щодо реалізації протоколу спілкування між репозиторіями. У секції, що стосується налаштування протоколів комунікації, наводиться лише загальний коментар, рекомендації застосовувати OAI протокол для доступу до цифрових бібліотек і приклад використання протоколу SOAP для передачі запитів та отримання даних. Проте досі залишається відкритим питання про достатність рекомендованого протоколу для вирішення складних задач із побудови такого репозиторію, що відповідає останнім вимогам адаптивного навчання [5].

Основним критерієм для добору протоколу є:

1. Легкість впровадження та реалізація справжньої архітектури відкритої мережі. Для репозиторіїв, що бажають якомога швидше стати інтероперабельними, наявність готових бібліотек під відповідну платформу.
2. Базування на XML як відкритому та безкоштовному стандарті, що не потребує жодного ліцензування.
3. Масштабованість, що забезпечує достатню ефективність за будь-якої кількості репозиторіїв у спільній мережі.
4. Сумісність зі SCORM - основним форматом НО у більшості наявних НС, таких як Moodle та ILIAS.

З точки зору кінцевого користувача, протокол має забезпечувати швидку та надійну передачу повнотекстових пошукових запитів. За свідчен-

нями розробників SPLASH [6], незважаючи на погодження всіх учасників проекту використувати спеціалізований пошук за метаданими для кращого результату, більшість користувачів здійснювали саме повнотекстовий пошук, хоча вже звикнувши до стереотипу пошуку у веб за допомогою Google [7].

Завданням протоколу eduSource Communication Layer є максимальна реалізація рекомендацій IMS DRI та застосування компонентів middleware для побудови мостів з іншими протоколами. Для того щоб зрозуміти строгі вимоги ECL щодо інтероперабельності, розглянемо ті типи мереж та спільнот, які беруться до уваги [8]:

Таблиця 1. Типи мереж та спільнот

| Вид мережі чи спільноти | Опис |
|--------------------------------|---|
| Серверні репозиторії | Більшість навчальних систем мають власні серверні сховища для збереження матеріалів з метаданими. ECL намагається охопити якомога більшу кількість таких сховищ, зокрема вирішити проблему об'єднання наявних сховищ, таких як CAREO. Проте нас цікавитимуть лише ті, що зберігають HO у форматі SCORM. |
| P2P клієнтські репозиторії | Проведений у 2003-2005 роках проект Lionshare дав змогу окремим користувачам створювати власні локальні сховища HO та обмінюватися ними. Проект не набув необхідної популярності, проте вказує на необхідність middleware для федералізації мереж репозиторіїв [9]. |
| Репозиторії метаданих | Ці репозиторії характеризуються спеціфікацією IMS DRI як <i>intermediary repositories</i> . Головне їх призначення - ведення реєстру наявних репозиторіїв та збирання метаданих для проведення ефективного пошуку. Такі сховища є альтернативою федералізованого пошуку - замість того щоб надсилати велику кількість запитів багатьом серверам, вони збирають метадані за результатами кожного пошуку, проводячи потім централізований запит по зібраних метаданих. Нові пошукові запити проводяться надалі лише за відсутності зібраної інформації. |
| Зовнішні репозиторії та мережі | eduSource приділяє велику увагу приєднанню інших наукових мереж та ініціатив. Це має дозволити користувачам здійснювати пошук поза межами мережі eduSource та навпаки - зовнішні репозиторії мають змогу приєднатись до загальної мережі. |

Наявність такої різноманітної аудиторії дозволяє розглянути принаймні три топології інтероперабельності: сітка (mesh), зірка та загальна шина доступу, які можна вирізнити за рівнем організованості та навантаженості.

Сітка є найбільш характерним та примітивним рівнем організації сховищ, при якому кожна пара учасників мережі фактично сама встановлює правила спілкування. Саме такий рівень організації зараз присутній у багатьох навчальних закладах України: більшість матеріалів передаються за домовленостями у вигляді відкритої медіа (гіпертекстові документи, малюнки, відео тощо), в архівах чи SCORM пакетах (імпорт та експорт відбуваються вручну чи напівавтоматично). На сьогодні практично кожна з НС дозволяє експортувати курс чи окремий пакет знань у вигляді архіву з метаданими, формат яких відрізняється. Проте недолік такого підходу очевидний - надмірна кількість транзакцій та низька ефективність при розповсюдженні пакетів HO на велику кількість учасників.

Топологія **зірки** дозволяє усунути проблеми попереднього рівня організації шляхом залучення центрального сервера, через який буде проходити обмін HO (забігаючи наперед, також зазначимо, що ця топологія частково використовується ECL). При використанні зіркової архітектури він працює як певний реєстр наявних учасників мережі (аналог реєстру сервісів UDDI), а також виступає у ролі посередника при здійсненні транзакцій. Сервер може служити сховищем метаданих, що описує наявні ресурси кожного з учасників для прискорення таких тривіальних операцій, як пошук, доставка та сповіщення. Але за даного розташування учасників центральний компонент зазнає значного навантаження, а у разі його відмови вся мережа стає недієздатною. Окрім того, для кожного з учасників порушується автономія спілкування - усі протоколи та формати даних визначаються централізовано.

Певним стандартом взаємодії систем сьогодні є топологія **загальної шини доступу**, що дозволяє створювати інтероперабельні мережі та підключати безліч учасників без значних витрат, а здешевлення доступу до Інтернет та підвищення пропускнуої здатності глобальних мереж заохочують до цього. Лише за останні півроку трафік найвідоміших соціальних мереж зріс на 25 відсотків [10].

Залежно від видів спільнот, що потребують доступу до загальної мережі на базі протоколу ECL, сформуємо ролі, функціональність яких має реалізовувати протокол інтероперабельності репозиторіїв HO: системний конфігуратор (відповідає за налаштування загальної інфраструктури мережі ECL), архітектор репозиторіїв навчальних матеріалів (зв'язує HO з необхідними метаданими, достатніми для пошуку та доставки HO), дизайнер (створює навчальні процеси згідно зі специфікаціями Learning Design [11] та

додає їх до пакетів НО), провайдер (сервіс чи експерт, що контролює електронні права DRM на час пошуку та доставки НО), шукач НО (простий користувач системи, як-от студент чи викладач, що шукає необхідну інформацію для реалізації власного навчального процесу). Кожен із цих видів користувачів володіє власним інструментарієм, що має доступ до ECL мережі чи конкретного репозиторію.

У результаті подальшого аналізу розглянемо загальну модель взаємодії компонентів та ролей системи (рис. 1):

Побудова загального протоколу дозволяє сформулювати основні компоненти відкритої мережі ECL: реєстр (має загальні метадані та інформацію про наявних учасників мережі), репозиторій (стандартизоване сховище НО з вбудованою підтримкою ECL), ECL шлюз (для з'єднання ECL репозиторіїв та інших сховищ даних та НС) та інші сховища навчальних матеріалів.

Як транспортний протокол ECL використовує SOAP без визначення складних типів. Оскільки нашим завданням є розширення протоколу для потреб з'єднання з репозиторіями, що, як правило, обмінюються SCORM пакетами, ми утримаємося від загального використання WSDL з його типізованою структурою, що унеможливорює необхідну гнучкість. Іншою причиною цього є використання асинхронних повідомлень, що необхідні для виконання таких функцій IMS DRI, як 'alert'. Наприклад, певному учаснику мережі ECL необхідно зібрати

курс, складові якого розміщені в різних репозиторіях, але в яких саме - невідомо. Отже, такий учасник надсилає до всіх відомих йому сховищ gather-запит. При появі у певному репозиторії наявної частини курсу учасникові-позивачу буде надіслано асинхронне повідомлення типу alert у вигляді SOAP пакета. Дізнавшись, звідки можна дістати певну частину курсу, учасник-агрегатор може надсилати подальші request-повідомлення для виконання збору НО. Іншим важливим аспектом використання саме асинхронних повідомлень є можливість організації P2P мережі - розподілений пошук по мережі вимагає саме асинхронного спілкування, оскільки виконання синхронних запитів чинить значне навантаження на шлюзи будь-якої P2P мережі. Подібним чином такий вид запитів ефективний при виконанні операції 'deliver' у разі, коли певному репозиторію необхідно доставити великий за обсягом пакет НО. Цілком розумно виконувати доставку кількома запитами, невеликими за розмірами, а в разі втрати частини пакета клієнт спокійно може виконати повторний запит на доставку.

Інша складова протоколу - власне стандартне повідомлення ECL, складається із заголовка та тіла:

- Message Header
- Message Type - власне, тип повідомлення, що відповідає опису сервісів. Додатково введено технічні типи CANCEL, ERROR, ACKNOWLEDGE.

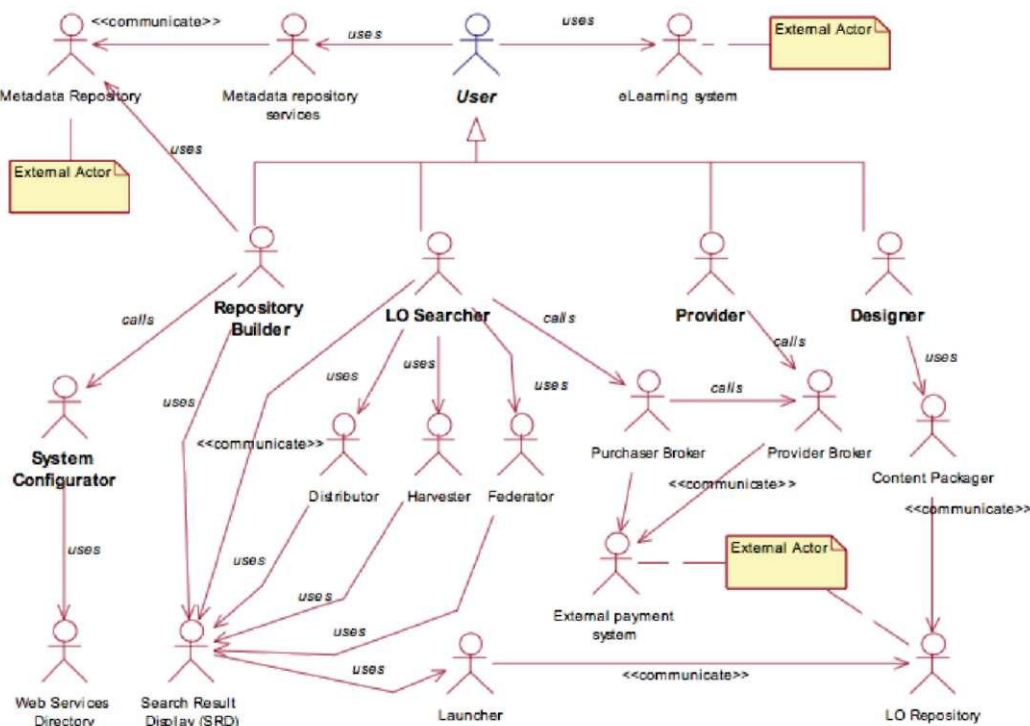


Рис. 1. Сценарії використання ECL протоколу

- **Communication Id** - унікальний ідентифікатор транзакції. Є однаковим для повідомлення-запиту та повідомлення-відповіді.
 - **Protocol** - хоча на даний момент єдиним протоколом є ECL, це поле стане у пригоді при розширенні можливостей.
 - **Version** - поточна версія протоколу, що використовується відправником. Поле необхідне для перевірки сумісності, а також реалізації зворотної сумісності з ранніми версіями ECL.
 - **Sender endpoint** - складна структура, що відповідає за місцезнаходження відправника повідомлення, якому отримувач має асинхронно надіслати відповідь. Приклади повідомлень із цією структурою розглянуто нижче. Структура також містить інформацію про те, яким методом буде надіслано зворотне повідомлення.
 - **Receiver service URN** - уніфіковане ім'я сервісу, за яким буде надіслано зворотне повідомлення.
 - **Error Message** - повідомлення про помилку в разі неуспішного виконання сервісу.
 - **Authentication** - закодована стрічка аутентифікації. Власне, розробники пропонують використовувати RSA для кодування і передавати відкритий ключ через реєстр репозиторіїв. Метод кодування може бути переглянутий залежно від вимог безпеки створюваного інтероперабельного репозиторію чи мережі репозиторіїв.
 - **Status** - використовується при асинхронній передачі повідомлень. Може набувати нульового значення (елемент відсутній), 'done' чи 'processing'. Останнє вимагає від репозиторію обробки повідомлення на надсилання відповіді, тоді як перші два стани свідчать про завершення транзакції.
 - **Message Body**
 - **Payload** - має у собі власне повідомлення ECL.
- На рис. 2 наведено приклад типового повідомлення ECL для виклику сервісу search. Серед загальної структури повідомлення слід звернути увагу на тіло запиту, що знаходиться всередині елементу <search>, пошуковий запит виконано за допомогою XQuery, проте цей метод може бути замінений і такими іншими методами, як здійснення безпосереднього пошуку елемента Xpath чи повнотекстового пошуку не лише серед метаданих, а й власне змісту НО. Відповідь на даний запит схожа за форматом, проте агрегує у собі ще метадані шуканих НО (у разі успіху, рис. 3).
- Складність ECL протоколу в реалізації може стати вирішальним фактором при його впровадженні. Саме тому більшість компонентів протоколу вже наявні у вигляді бінарних бібліотек java, зокрема eduSource Connector - бібліотека базових функцій репозиторію, що потребує від розробника лише реалізації потрібних методів, яка, проте, дає змогу під'єднатися до інших репозиторіїв для виконання, наприклад, пошуку чи доставки НО. Це дозволяє змінювати власне протокол без потреби внесення змін у клієнтські застосування - необхідно лише оновити версію бібліотеки ECL.
- Для забезпечення базових функцій спілкування клієнту мережі ECL необхідно реалізувати на вибір, відповідно до IMS DRI, такі інтерфейси: expose, gather, search, alert, submit, store, request, deliver. Найбільш пріоритетними сервісами для репозиторіїв, що присутні в НС типу ILIAS чи Moodle, є: search - можливість здійснення пошуку, оптимальним для даних репозиторіїв є використання XQuery, хоча можна розглядати і Xpath як більш просту альтернативу;

```

- <ecl xmlns="http://www.edusource.ca/xsd/ecl_v1p1">
- <header>
  <type>search</type>
  <cid>tui087E2FF1CC9C407E93ADA39DE7201625</cid>
  <protocol>ECL</protocol>
  <version>0.1</version>
  <receiver>search_service</receiver>
- <sender>
  <endpoint_type>SOAP_PULL</endpoint_type>
</sender>
</header>
- <body>
- <payload>
  - <search>
    <pattern>0</pattern>
    - <xquery>
      <results>{ FOR $lom in document("loms.xml")/lom WHERE contains($lom,
        "java program") RETURN {$lom} }</results>
    </xquery>
  </search>
</payload>
</body>
</ecl>

```

Рис. 2. Приклад повідомлення ECL.

```

- <ecl xmlns="http://www.edusource.ca/xsd/ecl_v1p1">
- <header>
  <type>expose</type>
  <cid>tui087E2FF1CC9C407E93ADA39DE7201625</cid>
  <protocol>ECL</protocol>
  <version>0.1</version>
  <receiver>search_service</receiver>
- <sender>
  <endpoint_type>SOAP_PULL</endpoint_type>
</sender>
</header>
- <body>
- <payload>
- <search>
  <pattern>0</pattern>
  - <xquery>
  - <results>
    + <lom xmlns="http://www.imsglobal.org/xsd/imsmd_v1p2"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_v1p2
        imsmd_v1p2p2.xsd">
    + <lom xmlns="http://www.imsglobal.org/xsd/imsmd_v1p2"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_v1p2
        imsmd_v1p2p2.xsd">
    </results>
  </xquery>
</search>
</payload>
</body>
</ecl>

```

Рис. 3. Приклад ECL повідомлення відповіді на стандартний запит пошуку

request - сервіс дозволу на скачування чи до- ставку знайденого НО типу SCORM; alert - мож- ливість асинхронного сповіщення клієнтських застосувань, що використовують НО репозито- рію; submit - сервіс розміщення чи оновлення НО у сховищі репозиторію.

В ідеальному світі розподіленої мережі на- вчальних матеріалів є лише один протокол та

єдиний спосіб пошуку, доставки й оповіщення. Проте наявна реальність - це гетерогенні мере- жі з наявністю великої кількості стандартів, ко- жен з яких домагається власної мети. Створен- ня розподіленої мережі ECL дозволяє наблизити ті реалії, коли будь-які знання можна вільно розповсюдити на відкритих інтелектуальних ринках.

1. Web 2.0 Wikipedia. Від 19 серпня 2008 року на http://en.wikipedia.org/wiki/Web_2.0.
2. Advanced Distributed Learning - SCORM. Від 20 серпня 2008 року на <http://www.adlnet.gov/scorm/>.
3. Amit P. S. Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics. Interoperating Geographic Information Systems. Springer, 1999.
4. IMS Global Learning Consortium: Digital Repositories Specification. - <http://www.imsglobal.org/digitalrepositories/>.
5. Aroyo L., Dolog P., Houben G-J., Kravcik M., Naeve A., Nilsson M. & Wild F. (2006). Interoperability in Personalized Adaptive Learning. Educational Technology & Society, 9 (2), 4-18.
6. Hatala M., Richards G. Making a SPLASH: A Heterogenous Peer-To-Peer Learning Object Repository. Proceeding of WWW 2003, May 20-24, 2003, Budapest, Hungary.
7. Google. - <http://www.google.com.ua/>
8. Hatala M., et al. The EduSource Communication Language: Implementing Open Network for Learning Repositories and Services. Proceedings of the 2004 ACM symposium on Applied computing, 2004, Nicosia, Cyprus.
9. LionShare: Connecting and Extending Peer-to-Peer Networks. - <http://lore.iat.sfu.ca/projects/lionshare.html>
10. Facebook Tops MySpace as Social Sites Globalize. Internet News. Від 6 вересня 2008 року на <http://www.internetnews.com/stats/article.php/3765406/Facebook+Tops+MySpace+as+Social+Sites+Globalize.htm>
11. IMS Global Learning Consortium: Learning Design Specification. Від 21 серпня 2008 року на <http://www.imsglobal.org/learningdesign/>

V. Yatsevsky

DEVELOPMENT OF ECL-BASED DISTRIBUTED LEARNING REPOSITORIES PROTOCOL

Defining the communication protocol in distributed environment of learning repositories is a primary task for building required infrastructure that would lead to effective exchange of learning objects. The article reflects basic requirements of the protocol and defines its common tasks that should be carried out. As a result of this work, a general configuration is suggested, based on the eduSource Communication Language (ECL).