

Цей консолідований інформаційний ресурс призначений для надання повної, достовірної інформації дебіторам про стан їхньої заборгованості.

1. Гуцак Є. *Особливості розвитку факторингових послуг в Україні* / Є. Гуцак, Ю. Шмагіна. – Режим доступу: [http://www.rusnauka.com/6\\_NITSHB\\_2011/Economics/3\\_80384.doc.htm](http://www.rusnauka.com/6_NITSHB_2011/Economics/3_80384.doc.htm). 2. Пальчук О. І. *Факторинг як специфічна фінансова послуга* / О. І. Пальчук. – 2010. – Режим доступу: <http://ena.lp.edu.ua:8080/bitstream/ntb/7839/2/28.pdf>. 3. Лисенко Ю. М. *Валютно-фінансовий механізм зовнішньоекономічної діяльності: навч. посіб.* / Ю. М. Лисенко. – К, 2005. – 179 с. 4. Руденко Л. В. *Розрахункові та кредитні операції у зовнішньоекономічній діяльності підприємства : підручник* / Л. В. Руденко. – К, 2002. – 258 с. 5. Белялов Т. Е. *Аналіз форм і методів управління дебіторською заборгованістю* / Т. Е. Белялов // *Актуальні проблеми економіки*. – 2005. – № 9. – С. 30–36. 6. Ткаченко А. М. *Використання факторингових операцій при управлінні дебіторською заборгованістю* / А. М. Ткаченко, М. В. Гугля // *Экономические проблемы и перспективы: сб. науч. тр.* – Донецьк, 2008. – Вип. 1. – С. 273–282. 7. Сорба А. І. *Сутність факторингу та динаміка його розвитку в Україні* / А. І. Сорба, В. Е. Любера, І. М. Літовська. – Режим доступу: <http://intkonf.org/sorba-ai-lyubera-ve-litovska-im-sutnist-factoringu-ta-dinamika-yogo-rozvitku-v-ukrayini/>.

УДК 621.38

О. Чкалов<sup>1</sup>, Б. Булах<sup>1</sup>, О. Безносик<sup>1</sup>, О. Мацулевич<sup>2</sup>

<sup>1</sup>Національний технічний університет України  
“Київський політехнічний інститут”,  
Навчально-науковий комплекс  
“Інститут прикладного системного аналізу”

<sup>2</sup>Таврійський державний агротехнологічний університет

## ОСОБЛИВОСТІ ФОРМУВАННЯ МАРШРУТІВ ПРОЕКТУВАННЯ В СИСТЕМІ GRIDALLTED

© Чкалов О., Булах Б., Безносик О., Мацулевич О., 2013

Розглянуто особливості формування маршрутів проектування в комплексі математичного моделювання GridALLTED. Наведено основні функції підсистеми формування маршрутів і потоків завдань, розглянуто формат опису маршруту завдання, що використовується для побудови та виконання потоків робіт за допомогою грід- та веб-сервісів. Показано напрям подальших досліджень.

**Ключові слова:** GridALLTED, моделювання, маршрут проектування.

This paper is devoted to the features of design route creation in the GridALLTED mathematical simulation complex. Task flow and route creation subsystem's main functions are given, the task route description format used for workflow building-up and executing by means of grid- and web-services is considered. The future investigation direction is denoted.

**Key words:** GridALLTED, simulation, design route.

### Вступ

З науково-технічним прогресом зростають вимоги до можливостей та швидкодії САПР. Новітні системи повинні розраховувати об'єкти, які складаються з сотень тисяч елементів, за максимально стислі строки, що неможливо здійснити на персональному комп'ютері чи потужному сервері. Зокрема, такі системи достатньо дорогі з фінансового погляду, що заважає використовувати їх в повсякденному житті рядовим користувачам. Ціни на деякі пакети моделювання можуть коливатися в межах від декількох тисяч до десятків тисяч доларів, що зменшує коло потенційних користувачів. Окрім цього, важливою проблемою під час використання програмного

забезпечення систем проектування є їх висока залежність від апаратної платформи, операційної системи, наявності додаткових програмних пакетів тощо. Особливою проблемою стає відсутність єдиного пакета, що дав би змогу виконувати моделювання в різних галузях. Виправити ці недоліки можливо завдяки створенню набору веб-сервісів, що надавали б віддалений доступ до різних програмних засобів через єдиний веб-інтерфейс, включаючи залучення потужних грід-ресурсів. Така система потребує механізму, який би забезпечував для виконання завдань зв'язок та передавання даних між різними модулями. Однією з таких систем є комплекс математичного моделювання GridALLTED.

### GridALLTED

Розроблення комплексу на основі сервісно-орієнтованої архітектури насамперед має на меті використання переваг сучасних веб- та грід-технологій при реалізації серверної частини на етапі розв'язання задач пошуку обчислювальних ресурсів, проміжного зберігання даних, забезпечення довготривалих розрахунків тощо.

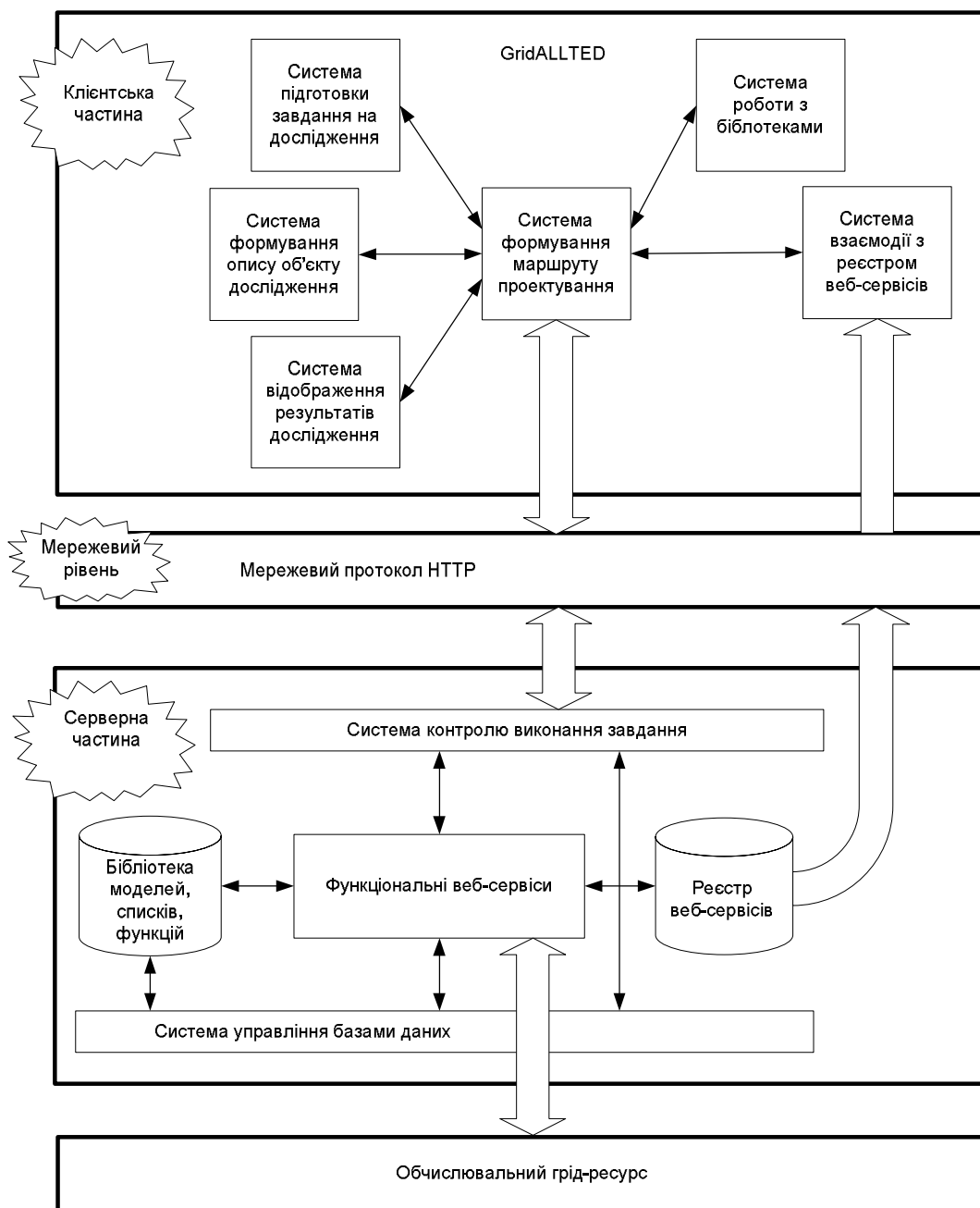


Рис. 1. Архітектура міждисциплінарного комплексу проектування GridALLTED

GridALLTED побудований на базі архітектури клієнт-сервер з використанням веб-сервісів (якими можуть слугувати й грід-сервіси, тобто веб-сервіси, що мають доступ до грід-середовища) на основі групи стандартів SOAP+WSDL+UDDI. Суть цієї архітектури полягає в тому, що весь обчислювальний процес ділять на набір незалежних веб-сервісів, що взаємодіють, що можуть бути розподілені поміж різними серверами (рис. 1). Доступ до веб-сервісів здійснюється через мережу із застосуванням стандартних протоколів типу HTTP. Як і у класичній архітектурі клієнт-сервер в цьому випадку можна виділити дві окремих частини системи – серверну та клієнтську [1, 2].

Серверна частина – це незалежна обчислювальна грід-система, що не тільки виконує роль обчислювальних потужностей для розрахунків, але й забезпечує пошук необхідних ресурсів для розв’язання задачі, організацію взаємодії між веб-сервісами та виступає системою масового обслуговування, що працює з клієнтськими задачами у багатозадачному режимі.

Як платформонезалежна клієнтська частини з інтерфейсом доступу до систем постановки, контролю виконання та відображення результатів завдання використовується звичайний браузер.

Клієнтська частина системи GridALLTED є інтерфейсом між “обчислювальною” частиною системи та користувачем. До вимог, що ставляться перед клієнтською частиною, можна зарахувати: виконання необхідних робіт з підготовки даних, потрібних для моделювання, у формі, зручній для користувача; виконання необхідних робіт з перетворення результатів моделювання у форму, зручну для перегляду та аналізу цих даних; виконання необхідних робіт над компонентами бібліотек моделей списків та функцій; дружній інтерфейс користувача; високу швидкість роботи в мережі; кросплатформність.

### Робочі потоки та маршрути проектування

Одним з головних елементів сучасних засобів організації числових експериментів є системи організації каскадів обчислень, відомі як workflow-системи [3, 4]. Такі системи (системи управління робочими потоками) дають змогу розробляти план (маршрут) обчислювального експерименту, динамічно коригувати його відповідно до цілей експерименту та його проміжних результатів, контролювати хід всього складного обчислювального процесу.

Під робочим потоком (або потоком/каскадом/маршрутом робіт) надалі в загальному випадку розумітимемо множину кроків, кожен з яких являє собою певну дію (обчислення), і переходів між ними, що організують виконання цих кроків у певній послідовності.

Відповідно до того, як визначена семантика переходів, розрізняють такі підвиди робочого потоку, як потік управління та потік даних. Реальний робочий потік є їх поєднанням.

Потік управління характеризує порядок виконання кроків потоку та організується за допомогою численних шаблонів, таких як розгалуження, паралельне виконання, умовний перехід, цикл тощо. Окрема увага може звертатися на обробку виключень та кроки з компенсації виконаних дій (це дозволяє визначити дії з відміни багатокрокової транзакції в разі помилок на будь-якому її етапі). У потоці, що позначає шляхи передавання даних, головна увага відводиться саме питанням передавання даних: копіювання, область видимості, узгодження форматів даних тощо (рис. 2).

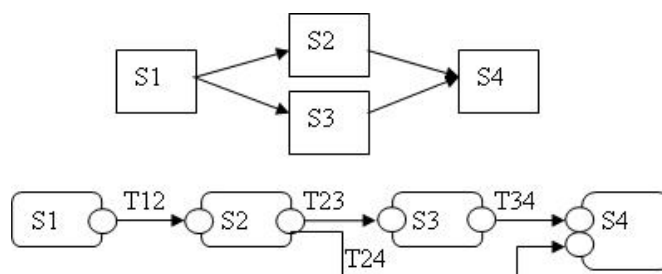


Рис. 2. Підвиди робочих потоків: потік управління (зверху, перший крок відіграє роль розгалуження або умовного переходу) та потік даних (знизу, позначено порти входних-вихідних даних)

Обчислювальними кроками потоку робіт можуть виступати кроки аналізу даних, їх обробки, процедури числового моделювання, оптимізації, візуалізації, що робить такий підхід придатним для реалізації інженерних та наукових систем аналізу даних та моделювання.

Як правило, побудова системи управління робочими потоками неможлива без формалізації потоку в межах деякої моделі та її опису конкретною мовою, в якій чітко визначено допустимі елементи потоків (з функціональної, поведінкової, інформаційної позицій) та їх семантику. Ні аналіз властивостей потоків, ні їх моделювання неможливі без прив'язки до конкретної моделі потоку, як і саме проектування потоків неможливе без прив'язки до мови, що описує вибрану модель.

Формалізація потоків, що фіксує їх семантику та допустимі елементи, є важливою умовою для створення автоматизованих середовищ проектування потоків, оскільки уможлиблює:

- автоматизоване проектування потоків з примітивів (елементів мови) у спеціалізованих графічних або текстових редакторах;
- автоматизований аналіз потоків на загальну ефективність, імовірність відмов, наявність конструктивних вад (моделювання);
- автоматизоване виконання створених потоків завдяки визначеній семантиці.

Застосування workflow-підходу у системах автоматизованого проектування може набувати форми “маршрутів проектування”, які складаються з набору операцій введення даних, математичних алгоритмів, обробки результатів тощо.

У межах створення міждисциплінарного комплексу математичного моделювання, який ґрунтується на використанні сервісно-орієнтованої архітектури, однією з ключових є система побудови маршруту обчислювального експерименту, або ж маршруту проектування. Аналогом подібних систем можуть бути системи побудови бізнес-процесів, але у випадку використання систем автоматизованого проектування побудова таких систем має певні особливості, як за складом елементів маршрутів, так і за взаємодією з іншими підсистемами комплексу.

Структура та функціональні можливості запропонованих підсистем комплексу націлені саме на формування “маршрутів проектування” [5], що дає змогу подати процес проектування у вигляді набору веб-сервісів, що взаємодіють, та дає можливість користувачу вибирати найдоцільніші шляхи розв'язання задачі.

Згідно із запропонованою концепцією користувачі комплексу можуть скласти і налаштувати потік робіт числового експерименту власноруч з наявних елементів (що представляють різні види аналізу, операції над математичними моделями тощо), а потім відправити його на автоматичне виконання.

Особливості маршруту, складеного з виконуваних модулів (веб-сервісів) для системи схематичного проектування, проявляються у тому, що певна послідовність дій можлива лише за умови виконання попередніх кроків, наприклад, формування завдання на дослідження здебільшого не має сенсу, якщо не виконано опис об'єкта дослідження тощо.

### **Підсистема формування маршрутів**

Підсистема формування маршрутів та потоків завдань призначена для візуалізації створення маршруту завдань у графічному інтерфейсі міждисциплінарного комплексу математичного моделювання GridALLTED з перетворенням його надалі на маршрут мовою опису XML та недопущенням помилок у коді опису маршруту ще на стадії його формування (рис. 3).

Основними функціями підсистеми формування маршрутів та потоків завдань є:

- генерація тексту опису об'єкта дослідження на основі візуально створеного маршруту;
- автоматичне створення потоку виконання;
- перевірка маршруту на недопустимі з'єднання.

Редактор маршрутів по суті являє собою “легкий” редактор напрямлених графів, який можна було б модульно використовувати як підсистему формування маршрутів та потоків завдань. Користувачу надається графічний редактор маршрутів, що уможлиблює побудову маршрутів проектування із зареєстрованих компонентів; склад бібліотеки компонентів маршруту може легко

розширюватись. Додавання елемента до маршруту здійснюється додаванням графічного представлення відповідного компонента та проведенням зв'язків для визначення послідовності проходження. За замовчуванням вибраний елемент маршруту є неактивним. Щоб активізувати елемент, треба ввести необхідні параметри, перелік яких залежить від вибраного елемента. Після того, як всі елементи, що входять до маршруту, будуть поєднані та активовані, стан маршруту переходить в "можливий до виконання". Створений маршрут зберігається у базі даних, прив'язаний до певного проекту користувача та може бути використаний повторно.

Складні, розгалужені маршрути передбачають паралельне виконання різних гілок маршруту, можливо, навіть на різних фізичних ресурсах. Отже, одночасно можна виконувати навіть ті проектні процедури, які зазвичай не можуть бути виконані послідовно, тобто для їх виконання потрібно було б скласти два незалежних маршрути та виконувати їх один за одним.

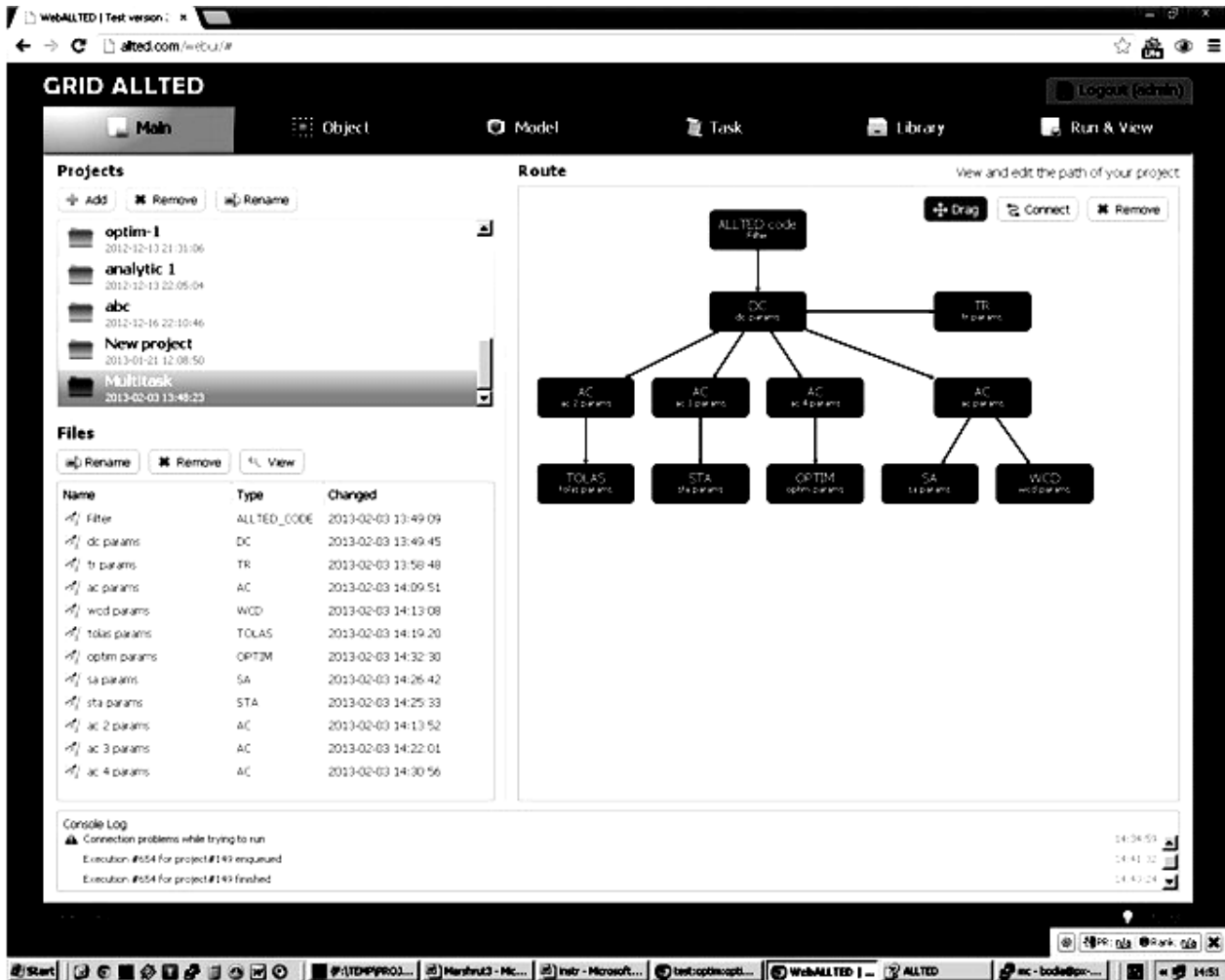


Рис. 3. Приклад маршруту обчислювального експерименту

### Формат опису завдання на виконання

Одним з важливих питань є формат обміну даними, який використовується для комунікації між шаром доступу (що представлений графічним редактором потоку робіт) і шаром виконання потоку робіт (що представлений системою управління потоками робіт). Є деякі загальні вимоги до такого формату:

- він має бути незалежним від платформи (тому що шари вищезгаданої архітектури можуть міститися на різних комп'ютерах з різною архітектурою, ОС тощо);
- він повинен бути незалежним від мови програмування (компоненти розподіленого комплексу можуть бути написані різними командами розробників різними мовами);

- він має бути сумісним із сервісно-орієнтованою архітектурою комплексу (а саме з SOAP веб-сервісами).

Очевидною відповіддю на вимоги, викладені вище, є використання розширеної мови розмітки (XML), оскільки це текстовий формат, незалежний від конкретної мови або платформи, і він безпосередньо використовується у стандартах, пов'язаних з веб-сервісами (SOAP, WSDL, BPEL – все це формати на основі XML).

Переваги використання мови XML такі: придатна для обробки машиною, водночас залишається зрозумілою для людини; нею можна описати складні структури даних (списки, дерева тощо); існує багато синтаксичних аналізаторів для більшості мов програмування. Також існує багато пов'язаних з XML стандартів, призначених для визначення структури XML (XSD), перетворення даних (XSLT) або запитів (XPath, XQuery).

Проте мова XML має й певні недоліки: її синтаксис є збитковим (XML-файли зазвичай більші, ніж бінарні чи навіть файли в інших текстових форматах, таких як JSON), вона не дуже придатна для передавання бінарних даних (вони повинні бути конвертовані в текст, а всі такі кодування, як base64Binary, HexBinary, призводять до збільшення розміру файла майже у два рази) тощо. Але, оскільки у САІР потік даних в основному складається з потоку документів, то мові XML можна віддати перевагу при виборі формату обміну даними [6, 7].

Під час реалізації комплексу GridALLTED було вирішено, що опис потоку робіт має бути чіткішим і лаконічнішим ніж WS-BPEL, який використовується для виконання потоків робіт. Отже, розроблено внутрішній простий XML-формат, що здатен як описувати потоки управління та даних, так і містити всі необхідні дані. Він оснований на декількох чільних концепціях потоку робіт, зображених на рис. 4 (пунктирні стрілки зображають потік даних, суцільні стрілки описують потік керування).

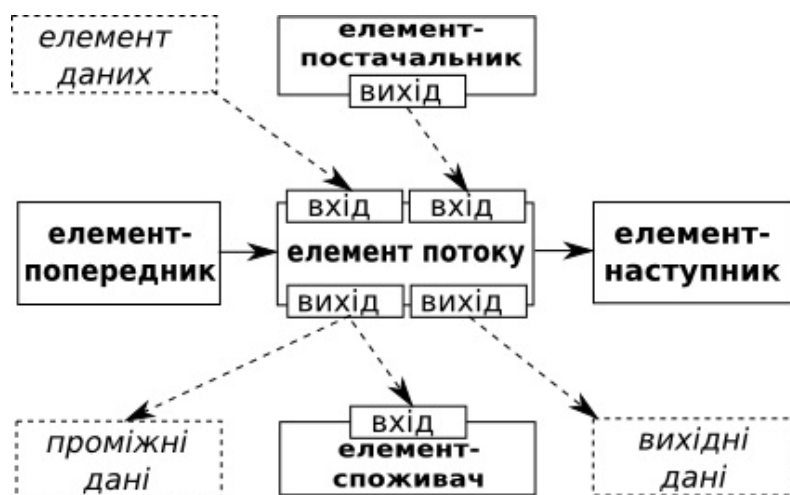


Рис.4. Основні сутності та типи взаємозв'язків у потоці робіт

Розроблений на основі XML формат опису завдання у вигляді потоку робіт [8] забезпечує уніфікований опис порядку обміну текстовими та бінарними даними, а також послідовність виконання дій у інженерних потоках робіт. Модель XML створюється на етапі запуску завдання на виконання на серверній стороні веб-сервісу. Цей файл містить всю необхідну інформацію про склад та конфігурації маршруту, що дає змогу однозначно інтерпретувати маршрут для його виконання в грід-середовищі.

Синтаксис формату опису потоку робіт, який використовується у GridALLTED, наведено нижче (“\*” означає “будь-яка кількість”, “?” означає “один необов'язковий елемент”, “+” означає “один або більше”):

```

<task>
  <workflow>
    <item
      id="item_id"
      name="item_type_name">+
    <after
      item="pred_item_id"/>*
    <input
      name="input_name">*
    <from
      item="source_item_id"?
      output="source_output_name"?
      element="data_element_id"?/>?
    </from/>
  </input>
  <output
    name="output_name"
    check="true"?
    return="true"?/>*
  </item>
</workflow>
<data>
  <input
    item="item_id"
    name="input_name"
    encoding="base64"?>*
    <![CDATA[input_data]]>
  </input>
  <element
    id="data_element_id"
    encoding="base64" ?>*
    <![CDATA[input_data]]>
  </element>
</data>
</task>

```

Кожен файл описує одне завдання (*task*). Опис завдання складається з двох частин: опису потоку робіт (*workflow*) та вхідних даних (*data*). Опис потоку робіт містить елементи потоку робіт (*item*) разом з їх взаємозв'язками: послідовністю виконання та порядком передавання даних. Кожен *item* являє собою екземпляр зареєстрованого елемента і має унікальний (для цього потоку робіт) ідентифікатор (атрибут *id*) та ім'я зареєстрованого елемента (*name*). Порядок виконання задається тегом *after*, який означає, що поточний елемент не може бути запущений до отримання результатів від елементів, ідентифікатори яких містяться в тегах *after* цього елемента. Для кожного елемента слід зазначити лише ті вхідні (*input*) та вихідні (*output*) порти, які необхідні для виконання цього потоку робіт, а їх імена (*name*) повинні відповідати іменам портів відповідних зареєстрованих елементів. Вхідні порти можуть містити тег *from*, який використовується для зазначення джерела даних. Якщо дані мають бути скопійовані з виходу іншого елемента, то атрибут *item* повинен містити ідентифікатор елемента-джерела, а атрибут *output* – ім'я виходу, з якого необхідно скопіювати дані. Якщо дані повинні бути скопійовані з блока *element*, то потрібно вказати лише атрибут *element* з ідентифікатором відповідного блока *element*. XML-тег *output* може мати кілька додаткових параметрів: *return* вказує на необхідність повернення даних з цього порту як частини вихідних даних потоку робіт, *check* вказує на можливість отримання даних з цього вихідного порту до закінчення всього потоку робіт (для контролю проміжних результатів). Блок даних може містити або вхідні дані для конкретних *item* потоку робіт (*input*), або елементи даних (*element*), які можуть бути використані декількома *item*. Бінарні дані повинні бути закодовані у base64, при цьому повинен бути встановлений відповідний атрибут (*encoding="base64"*).

Велику увагу під час створення підсистеми проектування маршрутів звернуто на розроблення правил, що визначають допустимість виконання сервісів у певній послідовності. Їх створює адміністратор системи при додаванні нових сервісів для забезпечення взаємозв'язку між модулями (приклад допустимих зв'язків між входами та виходами різних сервісів наведено в таблиці).

**Приклад допустимих зв'язків між входами та виходами різних сервісів**

Вхід/вихід	Сервіс 1	Сервіс 2	Сервіс ...	Сервіс N
Сервіс 1	X	+	-	+
Сервіс 2	+	X	+	-
Сервіс ...	+	-	X	+
Сервіс N	-	-	+	X

На практиці іноді виникає проблема включення в маршрут проектування нового функціонала сторонніх виробників, зумовлена неможливістю безпосередньо додати в програмний модуль представлення вхідних та вихідних даних у необхідних форматах через закритий вихідний код програми. У цьому випадку потрібно створити сервіс, що активуватиметься неявно (тобто графічний блок, що відповідає цьому сервісу, на схемі маршруту буде відсутній), який би містив правила перетворення вихідних даних одного модуля на формат даних входу іншого модуля.

Перспективним напрямом подальшого вдосконалення функціонала створення маршрутів проектування вважають можливість з'єднання різних гілок маршруту в деякий загальний компонент, при цьому подальше виконання маршруту має відбуватися на основі кращих результатів з отриманих на попередніх етапах.

**Висновки**

Представлення маршруту проектування у вигляді окремих функціональних сервісів доцільне у тих випадках, коли на сервіс покладається складна обчислювальна задача (стосовно часу виконання) або сервіс надає допоміжний (нереалізований у комплексі) функціонал (наприклад, у разі проведення міждисциплінарних досліджень), оскільки використання великої кількості “дрібних” сервісів значно збільшує час виконання.

Розроблений формат файла опису маршруту містить всю інформацію, необхідну для забезпечення послідовного (або, за потреби, паралельного) виконання окремих сервісів та обміну даними між ними та може бути наданий стороннім розробникам програмного забезпечення з метою реалізації інтерфейсів взаємодії їх програмних модулів з наявними сервісами.

Долучення до складу комплексу нових сервісів із закритим до зміни кодом можливе з використанням неявних (“службових”) сервісів, що мають забезпечити узгодження нових та наявних сервісів за вхідними та вихідними даними. Для недопущення конфліктів, пов'язаних з можливістю взаємодії різних сервісів у разі їх включення в маршрут проектування, нові сервіси необхідно додавати разом з правилами їх взаємодії з іншими сервісами.

1. Петренко А.І. Дослідження архітектури комплексу схемотехнічного моделювання GridALLTED / Петренко А.І., Ладогубець В.В., Фіногенов О.Д., Булах Б.В. // Вісник Університету “Україна” : Інформатика, обчислювальна техніка та кібернетика. – К. : Університет “Україна”, 2011. – № 2. – С. 65–70. 2. Петренко А.І. Архітектура мережевого комплексу схемотехнічного проектування ALLTED / Петренко А.І., Ладогубець В.В., Воевода О.О. // Комп'ютерні системи проектування. Теорія і практика: Вісник Нац. ун-ту “Львівська політехніка”. – 2005. – № 522. – С. 30–33. 3. Yu J. A Taxonomy of Workflow Management Systems for Grid Computing / Yu J., Buyya R. // Journal of Grid Computing. – 2005. – 3, N 3. – P. 171–200. 4. Taylor I.J. Workflows for e-Science. Scientific Workflows for Grids / I.J. Taylor, E. Deelman, D.B. Gannon, M. Shields (Eds.). – Springer. – 2007. – 530 p. 5. Petrenko A. Simulation in GridALLTED Complex / Anatolii Petrenko, Mykhailo Lobur, Volodymyr Ladogubets, Oleksii Finogenov, Bogdan Bulakh, Tetyana Ladogubets // The Experience of Designing and Application of CAD Systems in Microelectronics : 12-th Intern. Conf. “CADSM'2013”,



19–23 February 2013, Polyana-Svalyava (Zakarpattia), Ukraine : proc. – Lviv, 2013. – P. 422–424. 6. Denysyuk P., Teslyuk V., Khimich I., Farmaga I. XML application for microfluidic devices description. // Proc. of IXth International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics, February, 20–24, Polyana, Ukraine. – P. 567–569. 7. Kiselev G. Workflow Task Description Format for GridALLTED Complex / Gennadiy Kiselev, Bogdan Bulakh, Oleksandr Beznosyk, Vitalii Chekaliuk // The Experience of Designing and Application of CAD Systems in Microelectronics : 12-th Intern. Conf. “CADSM’2013”, 19–23 February 2013, Polyana-Svalyava (Zakarpattia), Ukraine : proc. – Lviv, 2013. – P. 425–426. 8. Лобур М.В. Управління процесом проектування в середовищі розподілених САПР / Лобур М.В., Лебедева О.О., Матвійків О.М. // Вісник Нац. ун-ту “Львівська політехніка”: Комп’ютерні системи проектування. Теорія і практика. – 2007. – № 591. – С. 16–21.

УДК 621.8

V. Lytvynenko

Kherson National Technical University,  
Dept. of Informatics & Computer Science

## SYNTHESIS OF THE WAVELET-NEURAL NETWORKS FOR THE CLASSIFICATION OF MASS SPECTRA USING CLONAL ALGORITHM

© Lytvynenko V., 2013

Мас-спектрометрію спектрів вибрано скринінговим засобом для виявлення дискримінаційних структур білка. Однак мас-спектри представляються даними великої розмірності, які мають значну кількість локальних максимумів (а.к.а. піки), що повинні бути проаналізовані. Для вирішення цієї проблеми ми розвинули нові триступеневі стратегії. Після попередньої обробки, для класифікації мас-спектрів, ми використовуємо алгоритм клонової селекції для синтезу колективних бінарних класифікаторів у вигляді вейвлет-нейронних мереж. Результати, отримані шляхом аналізу сукупності даних пухлини/здорові зразки, дозволили нам правильно класифікувати більш ніж 99 % проб.

**Ключові слова:** мас-спектр алгоритм, клональний вибір, бінарна класифікація, вейвлет нейронні мережі, MALDI-TOF , SELDI-TOF, MATLAB, WEKA

The mass spectrometry spectra are recognized as a screening tool for detecting discriminatory protein patterns. However, the mass spectra represent high dimensional data that have a large number of local maxima (a.k.a. peaks) which have to be analyzed; to tackle this problem we have developed a new three-step strategy. After preprocessing for classification of mass spectra, we use an algorithm clonal selection for synthesis collective binary classifiers in the form of wavelet-neural networks. The results obtained by the analysis of a data set of tumor/healthy samples allowed us to correctly classify more than 99% of samples.

**Key words:** mass spectra algorithm, clonal selection, binary classifiers, wavelet neural networks, MALDI-TOF , SELDI-TOF MATLAB, WEKA.

### 1. Introduction

SELDI-TOF (Surface-Enhanced Laser Desorption and Ionization Time-Of-Flight) technology is considered a modified form of MALDI-TOF (Matrix-Assisted Laser Desorption and Ionization Time-Of-Flight). According to these techniques, proteins are co-crystallized with UV-absorbing compounds, then a UV laser beam is used to vaporize the crystals, and ionized proteins are then accelerated in an electric field. The analysis is then completed by the TOF analyzer. The differences in the two technologies, which reside mainly in the sample preparation, make SELDI-TOF more reliable for biomarkers discovery and other