

КОНЦЕПЦІЇ ТА МОДЕЛІ ІНТЕГРАЦІЇ ОБ'ЄКТІВ МЕТАТЕХНОЛОГІЇ В СЕРЕДОВИЩІ METACASE

В.М. Зінькович, Є.І. Моренцов

Інститут програмних систем НАН України
03187, Київ, проспект Академіка Глушкова, 40
тел.: 526 42 86; e-mail: yevhen18@diawest.net.ua

Розглянуто підхід до інтеграції методів розробки систем та інструментальних засобів у метатехнологічному середовищі розробки. Викладено концепцію інтеграції процесів і концепцію представлення інформації в метатехнологічному репозитарії з використанням гетерогенних систем нотацій. Наведено концептуальні моделі інтеграції процесів і реалізації механізмів як інтеграції даних, так й інтеграції представлень через репозитарій.

The approach to integration of system development methods and tools in a metatechnological development environment is considered. The concept of processes integration and the concept of information representation in metatechnological repository with usage of heterogeneous notations systems are explained. Conceptual models of processes integration and of implementation of mechanisms both of data integration and of notations integration through repository are presented.

Вступ

Для ефективного створення великих та складних програмних систем (ПС) необхідно спочатку сформувати конкретну проблемно-орієнтовану технологію програмування (ПОТП), яка максимально враховує всі особливості ПрО. Формування технологій під вимоги реалізації ПС є основною задачею метатехнології — комп'ютеризованої технології вищого рівня, призначеної для побудови технологій програмування під конкретне застосування [1]. У метатехнології ПОТП представляється моделлю, яка містить чотири складових: модель життєвого циклу (процесу розробки ПС), методи розробки (МР) ПС, нотацій для представлення стану розроблюваної ПС та її компонентів у процесі розробки та інструментальні засоби (ІЗ), які реалізують методи й автоматизують процес розробки ПС.

Ці складові утворюють чотири відповідних класи об'єктів метатехнології, потрібні екземпляри яких поєднуються в єдину технологічну систему — ПОТП, яка повністю детермінує процес розробки визначеної ПС.

Узагальнений процес розробки ПС представляє собою сукупність процесів, які реалізують одну або декілька технологічних операцій, і порядок виконання процесів задається настановою процесу (можуть бути присутні як послідовні, так і паралельні гілки). Тією ж таки настановою визначаються: вхідні і вихідні дані процесу – початковий і кінцевий стан розроблюваного об'єкта, представлений відповідною нотацією, використовувани в процесах методи розробки та інструментальні засоби. Схематично узагальнений процес розробки та використовувани в ньому метатехнологічні об'єкти показані на рис. 1.

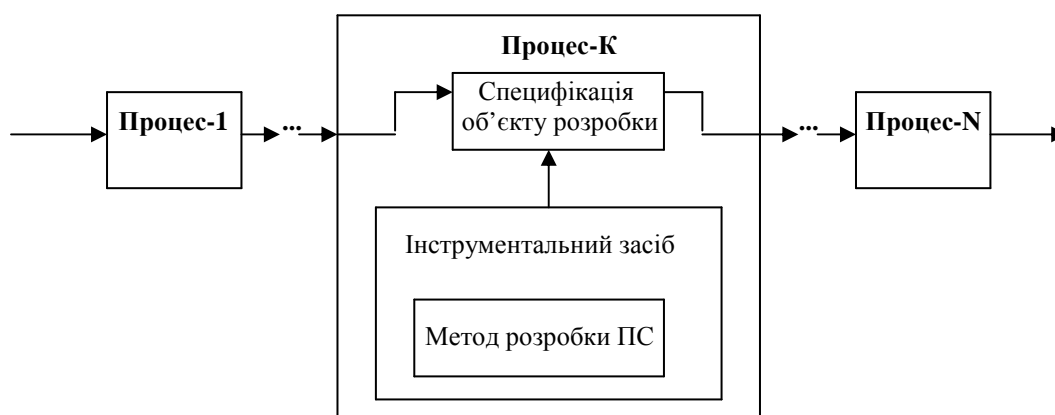


Рис. 1. Узагальнене представлення процесу розробки ПС

Таким чином при створенні ПОТП виникає задача інтеграції об'єктів різних класів у єдину технологічну систему. Ця задача вирішується на рівні метатехнології.

Інтеграція в метатехнології – це процес об'єднання об'єктів, компонентів метатехнології в повну систему типу MetaCASE. Інтеграція полягає у перетворенні набору множин незв'язаних метатехнологічних об'єктів, що включають інструменти, методи розробки, процеси і дані, у метатехнологічне середовище розробки (МСР) програмних систем. Застосовувані для цього механізми включають:

інтеграцію даних – спільне використання інформації усередині МСР через метатехнологічний репозитарій, загальнодоступний для всіх компонентів МСР;
інтеграцію управління – синхронізацію дій розробників у межах МСР;
інтеграцію представлень, що забезпечує однотипне представлення компонентів у МСР;
інтеграцію інструментальних засобів;
інтеграцію методів розробки.

При цьому два останніх механізми інтеграції у значній мірі реалізуються шляхом використання перших трьох механізмів.

Слід зазначити, що при створенні ПОТП задача інтеграції розглядається як задача інтеграції об'єктів одного класу на послідовності процесів. Задача інтеграції об'єктів різних класів, що може виникнути у метатехнології [2] вирішується поза процесом створення ПОТП, оскільки, як правило, ІЗ розробляється для конкретного МР, який використовує цілком конкретну нотацію для представлення даних.

Початок розробці концепції інтеграції у галузі програмної інженерії було покладено з виникненням перших CASE-систем. На першому етапі була потреба в інтеграції тільки CASE-інструментів. Із розроблених протягом останнього десятиріччя чотирьох видів інтеграції інструментів на даний момент загальноприйнятою стала концепція інтеграція CASE-інструментів у переносне загальне інструментальне середовище (в англійській аббревіатурі Portable Common Tool Environment — PCTE). На даний момент PCTE – фактично є загальним стандартом для інтерфейсу інструментів на основі відкритого репозитарію. Воно визначає набір загальних і стандартних послуг, призначених для підтримки переносних і таких, що мають здатність до інтеграції, інструментальних засобів CASE. Введення PCTE в структуру інтеграції інструмента дозволяє здійснювати простий обмін даними між інструментальними засобами з різними інтерфейсами за даними. PCTE реалізує схему, що дозволяє різним інструментальним засобам звертатися до загальної інформації в репозитарії, спільно використовувати її та управляти нею.

Крім PCTE існує стандарт ATIS (A Tool Integration Standard) та його більш сучасне найменування – CIS (Component Integration Standard), який являє собою пропозицію об'єктно-орієнтованого підходу до інтеграції інструментів, що забезпечує множину інтерфейсів, які підтримують схемо-керуєму диспетчеризацію поведіння. ATIS/CIS представлений у термінах ієрархії, що специфікують і взаємопов'язують абстрактні типи даних, їхні властивості та методи. Іншими словами, ATIS/CIS пропонує зразок об'єктно-орієнтованої моделі, що визначає множину класів та їхніх властивостей. Ця модель спроектована для забезпечення сервісів інструментів, що сприяють інтеграції і полегшуючих розробку інструментів.

Крім інтеграції ІЗ, дослідження проблеми інтеграції в CASE-системах також були сфокусовані на інтеграції об'єктів. Якщо інтеграція інструментів має справу з неявним звертанням й управлінням інструментальними засобами розробки, то інтеграція об'єктів забезпечує несуперечливе представлення артефактів розробки й простих користувальницьких інтерфейсів, щоб генерувати їх, звертатися до них і управляти ними [3, 4].

Однак із переходом досліджень на рівень метатехнології виникла потреба у принципово новому типі інтеграції, який би об'єднував два вищезазначені рівні інтеграції і, за можливістю, підтримував інтеграцію МР. Така сукупна інтеграція можлива при інтеграції процесів. Відправною точкою для введення такого типу інтеграції є те, що у процесі фактично поєднується “трійка”: об'єкт – метод – інструмент. Інтеграція процесів детермінує операції розробки ІЗ через модель процесу (МП) і забезпечує координацію та управління процесами, поряд із інтеграцією процесів і об'єктів. Головні вигоди від інтеграції процесів у технології розробки програмного забезпечення полягають у чітко детермінованому процесі управління і, відповідно, поліпшеному управлінні проектом. Інтеграція процесів — це процес, який за суттю є стратегією для створення процесо-центрованих MetaCASE-середовищ.

1. Концепція інтеграції процесів

Інтеграція інструментів забезпечує набір ІЗ для розробки програмного забезпечення і механізми звертання, що управляють їхнім використанням. Фактично, ці ІЗ формують концептуальний (динамічно налагоджуваний) ланцюжок звертань, який зв'язує операції процесу розробки ПС. Як правило, версія ланцюжка звертання залежить від ступеня гнучкості використовуваної технології розробки ПС, а ступінь змін у різних версіях ланцюжка звертань залежить від потужності набору ІЗ. Чим більший цей набір, тим більше різновидів може мати ланцюжок звертань в одній й тій же ПОТП.

Об'єктна інтеграція, заснована на об'єктній моделі програмних артефактів, підкреслює управління артефактами протягом розробки. Продукування і застосування цих артефактів зазвичай мають частковий порядок, тобто артефакт, вироблений на ранній операції життєвого циклу, буде використовуватися в наступних операціях життєвого циклу для створення іншого артефакту. Неформальний опис вимог до ПС, як правило, є початковим артефактом, а кінцевий артефакт – це готова ПС. При об'єктній інтеграції теж присутній концептуальний ланцюжок трансформації ресурсу, у якому початкові артефакти розробки послідовно перетворюються в проміжні, а потім – у кінцевий програмний продукт. З іншого боку, об'єктна інтеграція не містить дій, що продукують і використовують програмні артефакти, і, відповідно, не визначає способів їх виконання (методів розробки).

Загальною особливістю цих двох концептуальних ланцюжків є те, що вони описують ланцюжок виконання задачі розробки ПС з різних точок зору. Ланцюжок звертань до інструментів детермінує

використання ІЗ, водночас ланцюжок перетворень ресурсу представляє введення і виведення при виконанні задачі. Інтеграція процесів робить концептуальний ланцюжок виконання задачі розробки ПС точно визначеним, гнучким до змін і ефективним для розробки програмного забезпечення. Цей ланцюжок представляється моделлю процесу і використовується для досягнення і підтримки більшого ступеня інтегрованості в ПОТП.

Інтеграція процесів забезпечує доступ до функцій, що реалізуються компонентами МСР, використовуючи механізми, які підтримують: управління процесом, робочу область, звертання до інструментів і управління об'єктами. Ці механізми, включають модель процесу, монітор процесів, інтерфейси розроблювача і менеджера (рис. 2). Вони є ключовими компонентами процесо-центрованого метатехнологічного середовища розробки ПС. Інтеграція процесів також дає можливість контролювати і керувати процесом розробки.

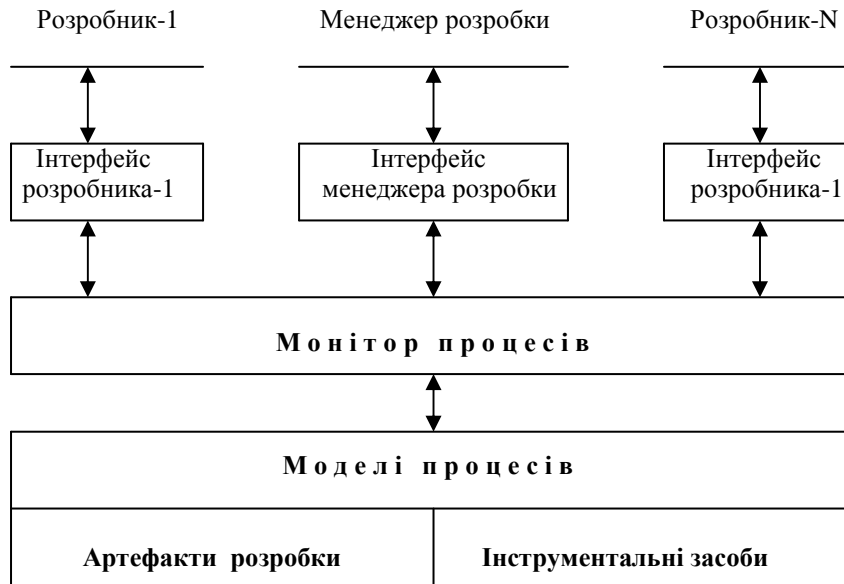


Рис. 2. Концептуальна модель інтеграції процесів

Модель процесу (МП) надає формальний спосіб організації й опису взаємодії моделі ЖЦ, МР, програмних артефактів, ІЗ і розроблювачів. МП визначена як набір об'єктів, у якості яких виступають операції розробки, МР, артефакти, ІЗ і розроблювачі. Кожний з цих об'єктів описує відповідну інформацію, яка використовується в процесі розробки ПС, і має власне представлення цієї інформації. Ці об'єкти зв'язуються через декілька видів відносин. У цілому, МП служить свого роду репозитарієм, у якому зберігається інформація про стан процесів розробки і дії (операції), що дає можливість керувати проектом розробки ПС.

Модель процесу визначає ієрархію дій, яка описує декомпозицію дій і операцій процесу розробки ПС і вимоги до ресурсів, включаючи програмні артефакти, інструментальні засоби, розроблювачів та інші критичні ресурси.

Ієрархія дій являє собою декомпозицію МП в ієрархію дій нижнього рівня, названих задачами й операціями. Рівні декомпозиції можуть бути довільними в залежності від складності процесу. Опис верхнього рівня в МП – це задача, що рекурсивно декомпозована в набір взаємодіючих підзадач (задачі й операції). Операції на нижніх рівнях цієї ієрархії являють собою єдиний виклик інструмента і простого перетворення ресурсу. У межах рівня декомпозиції зазначено також частковий порядок виконання підзадач. Порядок виконання визначає кілька типів відносин старшинства серед підзадач: послідовний, паралельний, ітераційний і умовний.

Чотири типи вимог до ресурсу визначають опис ресурсів, необхідних для підзадачі, й очікуваний результат. По-перше, це прив'язка розроблювачів до організаційних ролей, що вони виконують протягом виконання підзадачі. По-друге, це програмні артефакти, що необхідні для виконання підзадачі, а також ті, котрі створюються або розширюються в процесі її виконання. По-третє, це використовувані ІЗ. І по-четверте, це інформація про планування підзадачі й про очікувану тривалість її виконання. Ці ресурси представлені як незалежні класи об'єктів і мають відносини, що зв'язують їх у МП.

Монітор процесів запускає виконання або інтерпретацію МП відповідно до ієрархії дій. Він керує порядком виконання підзадач і обмеженнями, які повинні бути задоволені до того, як підзадачі будуть виконані. Монітор процесів фактично відіграє роль автоматизованого монітора, що ініціює МП; визначає виконувані підзадачі; приймає вхідну інформацію від розроблювачів; модифікує стан цих підзадач; якщо треба, то розмножує їх, і потім викликає інші підзадачі.

Ключова перемінна, котра визначає стан виклику процесу, – його статус. Статус МП або підзадачі – це індикатор її поточного стану в процесі розробки. Стан модифікується монітором процесів на підставі його взаємодії з розроблювачами. Зміна поточного значення статусу означає виконання виклику. Статус дії

визначається її складовими операціями, у той час як статус задачі визначається рекурсивно на статусах складових підзадач. Таким чином, у той час як МП вказує на запропонований порядок виклику підзадач, запис переходів стану підзадачі визначає порядок, у якому фактично викликалися підзадачі.

В узагальненому вигляді процес розробки ПС виглядає наступним чином. Як тільки розподіл ресурсів зроблений, розроблювачі стартують виконання розподіленої МП через монітор процесів. При виконанні дій (операцій) деякі з них можуть бути виконані успішно, інші можуть бути припинені, виконання третіх може бути порушено через помилки і вони мають потребу в відновленні або перевпорядкуванні дій, для їхнього повторного виконання. Після того, як визначені дії (операції) виконані, вони звертаються до монітора процесів, щоб модифікувати МП й ініціювати виконання подальших дій. Виконання дій продовжується поки МП не виконана цілком. Відповідно, розроблювачі управляють МП через два різних користувальницьких інтерфейси.

Інтерфейс розроблювача – це робоче середовище, що установлене МП. Цей інтерфейс дає можливість різним розроблювачам виконувати МП водночас й при цьому виконувати тільки призначені саме їм підзадачі через настанову процесу і робоче середовище.

Настанова процесу – це специфікація розподілу підзадач між розроблювачами і моменту початку виконання цих підзадач. Інтерфейс розроблювача підтримується настановою процесу через навігацію у межах МП за допомогою попередньо описаного монітора процесів. Інтерфейс розроблювача забезпечує початок виконання підзадачі тільки з ініціативи призначеного їй розроблювача, якщо її статус «Готово» чи «Зупинено». Коли підзадача закінчена, інтерфейс розроблювача інформує монітор процесів, який встановлює підзадачу-наступник у стан готовності до виконання, після чого вона може бути розпочата призначеними їй розроблювачами. Таким шляхом розроблювач може запустити призначену йому задачу (підзадачу), як тільки вона отримує статус готовності й задоволені всі вимоги до ресурсів, одразу або через деякий проміжок часу.

Робоче середовище. Створення робочого середовища включає два аспекти: встановлення робочого середовища і виклик необхідних ІЗ з ресурсами як параметрами. Робоче середовище в інтерфейсі розроблювача – це середовище, в якому він може виконати дію (операцію). Робоче середовище формується на основі специфікації дії. Ця специфікація, забезпечена МП, включає необхідні для дії ресурси, забезпечені ресурси та ІЗ. Це дає можливість викликати ІЗ тільки за наявності забезпечених для них ресурсів. У простій ситуації, коли у дії (операції) визначені тільки один інструмент і один ресурс, виклик інструмента виконується автоматично, як тільки дія отримує статус «Готово».

Інтерфейс менеджера процесу забезпечує менеджерів і аналітиків процесу ІЗ, які визначають, відображають і управляють МП, та можуть виконуватися водночас з інтерфейсом розроблювача.

З погляду монітора процесів, інтерфейс менеджера не змінює стан МП, але відшукує інформацію в МП і представляє її в графах і таблицях. Ці графи і таблиці також модифікуються монітором процесів у реальному часі, як тільки успішно завершилася призначена задача.

Відстеження виконання процесу включає спостереження за просуванням процесу: завершення підзадачі, створення ресурсу, дії розроблювачів. Контролювання процесу також включає зміну значень у МП. Для цього використовуються можливості призначення задач розроблювачам, видалення розроблювача, розподілу ресурсів.

Представлена концепція інтеграції процесів являє собою основу стратегії створення процесо-центрованого середовища MetaCASE. Слід зазначити, що інтеграція процесів підтримує архітектуру відкритої системи, що дозволяє поміщати різні ПОТП у процесо-центроване MetaCASE-середовище.

2. Особливості інтеграції методів. Концепція інтеграції даних та представлень

Методи розробки систем (в англійській аббревіатурі System Development Methods – SDM) реалізуються через інструментальні засоби. Іншими словами МР визначають ІЗ для представлення стану розроблювальної ПС за допомогою системи нотацій. Оскільки у метатехнологічному середовищі ІЗ фактично є відображенням МР, то вирішення задачі інтеграції ІЗ через концепцію інтеграції процесів частково вирішує задачу інтеграції як ІЗ, так і МР, принаймні, інтеграцію по управлінню. Водночас повне вирішення задачі інтеграції можливе лише шляхом застосування вищезазначених механізмів інтеграції даних та інтеграції представлення. Механізм інтеграції даних забезпечується спільним використанням інформації всередині МСР через метатехнологічний репозитарій, загальнодоступний для всіх компонентів МСР, а механізм інтеграції представлення – однотипним представленням компонентів у МСР.

Щоб спільно використовувати дані, МР та ІЗ повинні вміти обмінюватися даними. Проблема тут полягає у тому, що дані, експортованим одним ІЗ/МР, необхідно конвертувати у формат імпорту іншого ІЗ/МР. Одним із способів рішення цієї проблеми з мінімальним числом конверторів є визначення канонічного формату для конвертування даних і використання конверторів для перетворення даних “експортний формат – канонічний формат – імпортний формат”. Основна проблема при цьому полягає у тому, що необхідно мати досить багату модель даних для представлення спільно використовуваних даних і одержання незалежної від конкретного інструмента інформаційної моделі для об'єктів, які інструментальні засоби використовують спільно, а також конвертори, що належним чином інтерпретують семантику конвертованих даних. Приклади такого підходу до обміну даними наведені в [5, 6]. Але такий підхід не базується на репозитарії.

Водночас, використання метатехнологічного репозитарію для збереження даних, спільно використовуваних інструментальними засобами, має три важливі переваги:

- 1) об'єкти зберігаються в репозитарії, а не розсіяні у файлах, що підтримуються різними інструментальними засобами;
- 2) існує тільки одна копія кожного спільно використовуваного об'єкта, що дозволяє уникати непогодженості між копіями того ж самого об'єкта, керованими різними інструментальними засобами;
- 3) не губиться інформація, що передається від одного інструмента до іншого.

Саме ці особливості зберігання об'єктів у репозитарії дозволяють запропонувати підхід до обміну даними з використанням так названого віртуального інтерфейсу репозитарія. Цей механізм передбачає, що вся інформація зберігається у репозитарії у чітко визначеній канонічній формі, а всі доступи інструментів до даних (з використанням інтерфейсу доступу до даних, властивому інструменту) перехоплюються і транслюються у доступи до об'єктів репозитарія. Цей механізм частково нагадує загальний спосіб інтеграції немодифікованих інструментальних засобів UNIX: всі операції з файлами в UNIX перехоплюються і направляються до репозитарія.

Застосування такого підходу потребує визначення засобів (нотацій) представлення канонічної інформаційної моделі. Практично зробити це в даний час дуже важко, тому що наявні міжнародні стандарти є неповними і немає домінуючого розроблювача, що диктує стандарт. Однак, в області CASE, підтримується розвиток CASE Data Interchange Format (CDIF), що є інформаційною моделлю для даних CASE. Крім того існує стандарт інтеграції інструмента ATIS (A Tool Integration Standard), що являє собою розширювану об'єктно-орієнтовану інформаційну модель, доповнену прикладним програмним інтерфейсом (API) для звертання до об'єктів репозитарія.

Водночас ми пропонуємо для представлення інформації у канонічній інформаційній моделі використовувати підхід, заснований на гетерогенній системі нотацій (ГСН) [7]. Ця система являє собою комбінацію нотацій, кожна з яких використовується відповідним інструментом (методом). За допомогою ГСН представляються гетерогенні специфікації – композиції часткових специфікацій, представлених у двох або більше системах нотацій. Потрібні композиції залежать від контексту і використовуваних ІЗ (МР) систем нотацій. На першому етапі при додаванні в МСР ІЗ (МР) використовується ним система нотацій просто додається до вже наявних. При цьому визначається алгоритм конвертації між системами нотацій, який потім використовується універсальним конвертором репозитарія. В подальшому, на основі спільних або аналогічних компонентів використовуваних систем нотацій створюється базис ГСН, і при додаванні нового ІЗ (МР) з його системи нотацій беруться лише компоненти, які відсутні в базисі ГСН, який розширюється також шляхом створення композицій компонентів систем нотацій на основі їх аналізу. Поточний стан ГСН та її базису представляється метамоделлю, яка фактично задає алгоритми конвертації специфікацій представлених у різних системах нотацій, що входять до ГСН. Цей алгоритм розширюється одночасно з нарощуванням ГСН.

При такому підході звертання будь якого інструмента (методу) за інформацією до репозитарію перехоплюється універсальним конвертором, який реалізовує механізм віртуального інтерфейсу репозитарія. Цей конвертор на підставі метамоделі ГСН виконує конвертацію запитуваної інформації у формат імпорту ІЗ. При розміщенні ІЗ результатів своєї роботи в репозитарії експортні дані не конвертуються, а представляються за допомогою системи нотацій інструмента. Виграш при цьому досягається на відсутності конвертації експортних даних і суттєвому спрощенні конвертації даних із репозитарію у формат імпорту ІЗ за рахунок виявлення і застосування спільних компонентів використовуваних систем нотацій. Концептуальна модель реалізації даного підходу показана на рис. 3.

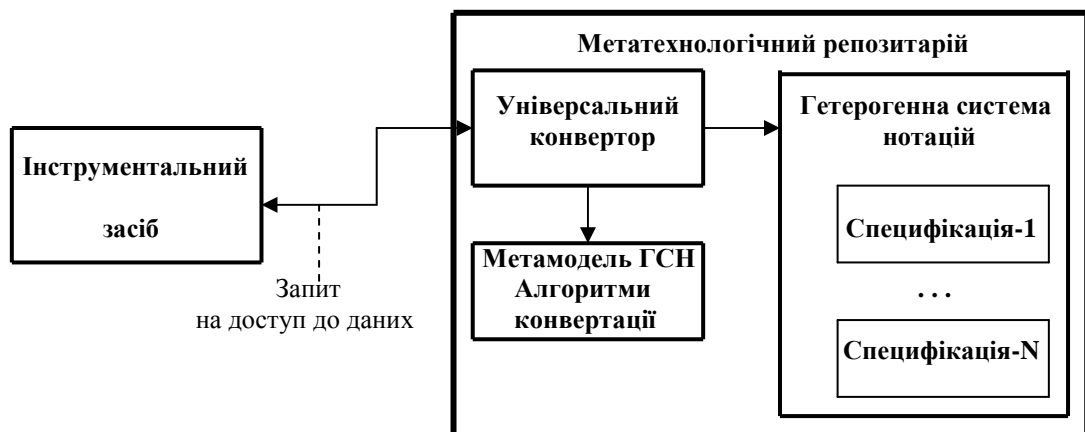


Рис. 3. Концептуальна модель реалізації механізмів інтеграції даних та інтеграції представлень

Висновки

Таким чином для інтеграції метатехнологічних об'єктів у середовищі MetaCASE ми пропонуємо використовувати механізми інтеграції даних, інтеграції управління та інтеграції представлень, а також концепції інтеграції процесів і представлення інформації про розроблювану ПС в репозитарії з використанням гетерогенної системи нотацій, які є базовими для реалізації зазначених механізмів. Реалізація цих механізмів

базується на використанні універсального конвертора та розширюваної метамоделі ГСН, яка містить алгоритми конвертації специфікацій.

Викладені результати отримані при виконанні фундаментального дослідження НАН України за темою “Дослідження та розробка метамодельного підходу до створення метатехнологічних систем проектування складних проблемно-орієнтованих програмних систем”.

1. *Зінкович В.М., Моренцов С.І. Концепції та засоби метатехнології для створення проблемно-орієнтованих технологій програмування // Проблемы программирования. – 2002. – № 3-4. – С. 74–82*
2. *Гринина М.А., Моренцов Е.И. Концепция интеграции методов, технологических и инструментальных средств в специализированных технологиях программирования // Там же.– 1997.– Вып.2. – С. 73–78.*
3. *Adams E.W., Honda M. and Miller T.C. Object Management in a CASE Environment. In Proc. of the 11th International Conference on Software Engineering. – Pittsburgh, PA. 1989, May. – P. 154–165.*
4. *Choi S.C. and Scacchi W. SOFTMAN: An Environment for Forward and Reverse CASE. Information and Software Technology, №33(9), 1991. – P. 78–95.*
5. *Product Data Representation and Exchange. STEP Part 21: Clear Text Encoding of the Exchange Structure (Physical File), ISO CD 10303-21, International Organization for Standardization, 1993.*
6. *Ring K. Start-Up Software One Claims to Have Nuts and Bolts of AD/Cycle, Computergram International, Issue No. 1512 (Sept. 14, 1990), Applied Data Services, England.*
7. *Richard F. Heterogeneous Notations for Pure Formal Method Integration. http://www-users.cs.york.ac.uk/~paige/Writing/pfm_paper.pdf*