

УДК 04.004

К.М. ЛАВРІЩЕВА

Інститут програмних систем Національної академії наук України
просп. Академіка Глушкова, 40, Київ, 03187, Україна

РОЗВИТОК ІДЕЙ АКАДЕМІКА В.М. ГЛУШКОВА З ПИТАНЬ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

Статтю присвячено 90-річчю від дня народження академіка НАН України і АН СРСР Віктора Михайловича Глушкова, його внеску в розвиток інформаційних технологій і технологій програмування з лініями виробництва програмних продуктів. Ще в 70-ті роки минулого століття В.М. Глушков вважав, що незабаром з'являться фабрики програм, які працюватимуть за принципом збирання, як в автомобільній промисловості, і з часом його передбачення повністю виправдалися. У статті в історичній послідовності розглянуто етапи розвитку технології програмування, приділено увагу створенню за концепцією В.М. Глушкова першої студентської фабрики програм та окреслено перспективи подальшого розвитку цього напрямку.

Ключові слова: комп'ютерна технологія, системна технологія, інформаційна технологія, технологія програмування, програмна інженерія, інформатика, збиральний конвеєр, методологія збирання.

*Присвячується 90-річчю
від дня народження
академіка В.М. Глушкова*

ВСТУП

Академік Віктор Михайлович Глушков, який усе своє життя присвятив створенню вітчизняної кібернетичної школи, залишив нам кілька парадигм, що стали основою для формування нових потужних наукових напрямів, а також багато концептуальних ідей, які великою мірою визначили подальший розвиток кібернетики. Його учні, Ю.В. Капітонова і О.А. Летичевський, у своїй книзі [1] сформулювали сім парадигм академіка В.М. Глушкова, коротко описавши їхню сутність. Більшість із них було реалізовано або пройшло апробацію ще за життя вченого, а деякі чекають свого часу й сьогодні.

Проте, на нашу думку, до числа парадигм В.М. Глушкова потрібно додати ще одну —

технологію програмування з лініями виробництва програм, яку він визначив у 1975 р. як перспективний спосіб для поступового переходу від ремісничого виробництва до промислового випуску комп'ютерів і програм [2–4]. В.М. Глушков зазначав, що індустрія програм має ґрунтуватися на технологічних лініях конвеєрного виготовлення програмних продуктів. Технології комп'ютерів, інформаційних і програмних систем він вважав рушійною силою прогресу фундаментальних кібернетичних наук, спрямованою на створення ЕОМ, їх системного забезпечення і вдосконалення схемної інтерпретації внутрішньої мови до рівня мови програмування, побудову мереж обчислювальних центрів і прикладного математичного забезпечення автоматизованих систем.

Нині технології програмування динамічно розвиваються від програмування й реалі-

зації задач проектування обчислювальних систем до побудови великих і малих комп'ютерних фреймворків, систем та кластерів. Огляд динаміки розвитку технологій програмування ми почнемо з періоду виникнення комп'ютерних та інформаційних технологій і програмних систем, простежимо етапи вдосконалення індустрії програмних продуктів завдяки інтеграції готових ресурсів і збиранню їх на фабриках програм, а також приділимо увагу сучасним методикам викладання курсів з програмної інженерії, технології програмування для студентів вищих навчальних закладів.

ПЕРІОД РОЗВИТКУ МОВ ПРОГРАМУВАННЯ (1958–1975)

Технології програмування почали розвиватися з часу розроблення мов програмування для розв'язання задач засобами ЕОМ за допомогою відповідних трансляторів і систем програмування. Цей період в Україні пов'язаний з активною діяльністю В.М. Глушкова зі створення ЕОМ та систем автоматизації програмування [1–7]. Загалом у цьому напрямі працювали спеціалісти з багатьох науково-дослідних інститутів СРСР.

Мови і транслятори. Перші транслятори, або програмуючі програми, ще в 50-х роках ХХ ст. розробили Е.З. Любимський, А.П. Єршов, М.Р. Шура-Бура для ЕОМ «Стрела» і «БЭСМ». Головними механізмами опису первинних програм для їхніх трансляторів були граф-схеми Янова і Ляпунова.

В Обчислювальному центрі АН УРСР, від самого його створення В.М. Глушковым у 1958 р., було взято проголошений академіком С.О. Лебедевим курс на побудову ЕОМ («СЭСМ», М-20, «Киев», «Промінь», «Днепр» та ін.) і створення для них систем автоматизації програм. У 1957 р. Віктор Михайлович запропонував метод автоматизації програмування, орієнтований на побудову програмуючих і бібліотечних програм. Апробував цей метод на прикладі систем диференціальних рівнянь його перший аспірант А.О. Стогній.

У той час головним програмістом ОЦ була Катерина Логвинівна Ющенко. Для

представлення програмуючих програм вона запропонувала одну з перших у світі мов програмування високого рівня — адресну мову, яку її співробітники Т.О. Крайнова, Л.П. Бабенко, Н.М. Грищенко та інші реалізували для опису трансляторів і програм на вітчизняних ЕОМ.

В Інституті кібернетики під керівництвом В.М. Глушкова почали розвивати такі основні напрями:

- технології побудови ЕОМ та обчислювальних комп'ютерів зі схемною інтерпретацією мов («Мир», «Україна»);
- автоматизація програмування (транслятори, операційні системи, налагоджувачі, редактори);
- технології побудови АСУ, систем автоматизації складних програм та пакетів прикладних програм різного призначення (математичного, економічного, транспортного, будівельного тощо).

На початку 60-х років розпочалося формування мов системного програмування для опису трансляторів і ОС. Після появи універсальної мови Алгол-60 у багатьох інститутах СРСР на її основі було створено транслятори: Т-1 (С.С. Лаврова, 1960); Т-2 (М.Р. Шура-Бура, Е.З. Любимський, 1961); Т-3 Альфа — російська версія Алголу (А.П. Єршов, 1963); Т-4 (К.Л. Ющенко, К.М. Лавріщева, 1965), розроблений для нової ЕОМ «Днепр-2» [8]. Для цих трансляторів та ОС спочатку використовували мову, близьку до внутрішньої мови ЕОМ, і варіант адресної мови для Т-4. Потім група Е.З. Любимського розробила проміжну мову АЛМО для трансляції з різних мов у код будь-якої ЕОМ [9]. Водночас спеціаліст з ОС В.П. Іванніков сформулював основи мови системного програмування для реалізації ОС на ЕОМ «БЭСМ» [10]. У той самий час за кордоном розробляли транслятори з мов Фортран (J. Backus, 1957), Алгол-60 (P. Naur, 1963) та ін.

Усі ці питання обговорювали на Першій (1965) та Другій (1968) Всесоюзних конференціях з програмування, які проходили в Києві [11], а також на багатьох наступних всесоюзних конференціях із цієї тематики,



Академік Віктор Михайлович Глушков

що відбувалися в різних регіонах СРСР аж до 1993 р. Ініціаторами цих наукових заходів були державні органи — ДКНТ, ДКНТІ СРСР — та інститути АН СРСР. На конференціях розглядали нові теоретичні й прикладні аспекти проектування, розроблення і тестування різних систем на великих ЕОМ, проблеми експлуатації та супроводження програмного забезпечення, створення інформаційних систем і технологій, а також методи індустрії програмної продукції, що стимулювало розвиток технології програмування.

Первинними мовами програмування були такі: адресна [12], алгебраїчний стиль програмування [6], конкретизовані граф-схеми Ляпунова і Янова в теоретичному програмуванні [13], синтезоване програмування [14], графічне програмування Р-метамовою [15], модульне програмування [16], блочне модульне програмування [17], композиційне програмування функціями і даними [18], програмування задач чисельного аналізу [19] та ін.

Отже, у програмуванні на перших ЕОМ в Україні використовували адресну мову — найпростішу для кодування програм. Потім з'явилися такі мови, як Алгол-60, Фортран, Кобол, Пролог, Ада та ін., проте вони не мали механізмів подання програм у наочній

і конструктивній формі. Тому алгоритми різних програм подавали за допомогою графових схем Ляпунова і Янова, нової Р-графічної мови, розробленої В.М. Глушковым та І.В. Вельбицьким, і структурної схематології в програмуванні й теорії автоматів і алгоритмів.

Теорія програмування. В.М. Глушков, як алгебраїст і кібернетик, дав формалізований опис теорії автоматів, алгоритмів, кібернетики, систем АСУ тощо [20, 21]; ініціював розвиток алгебраїчного програмування із застосуванням математики, теорії автоматів і доказу теорем; запропонував алгоритмічну мову АНАЛІТИК для опису алгебраїчних перетворень, використану на ЕОМ «Мир» [6]. Разом із завідувачем відділу К.Л. Ющенко і аспірантом Г.О. Цейтліним він розробив теоретичні аспекти програмування на основі адресної мови, концепцію синтаксичного і семантичного аналізу в трансляторах для різних мов, а також систему алгоритмічних алгебр на основі теорії автоматів [12]. Упродовж багатьох років Г.О. Цейтлін розвивав систему алгоритмічних алгебр, реалізував її в системі Мультипроцесист, а в останнє десятиріччя цю теорію було доповнено засобами генерації і паралельного програмування, зокрема аспіранткою О.А. Яценко [22].

Автоматизація програмування. Одним із перших засобів автоматизації програмування була розроблена Е.Х. Тиугу система ПРИЗ, основана на методі синтезу програм із понять, описаних на PL/1, Фортрані, Асемблері [14]. У ній започатковано новий метод — синтезувальне програмування, яке стало інструментом побудови моделей пакетів прикладних програм із вхідною мовою їх опису «Утопіст».

Алгебраїчне програмування набуло розвитку під час створення під керівництвом В.М. Глушкова серії машин «Мир» для інженерних розрахунків, для якої було розроблено алгоритмічну мову АНАЛІТИК, орієнтовану на опис і реалізацію чисельно-аналітичних методів. Вона виявилася ефективним інструментом для конструювання формул, їх оптимізації та обчислень похідних, диферен-

ціальних рівнянь високих порядків, кратних інтегралів тощо. У 60-х роках роботи зі створення математичних моделей для дослідження технічних і комп'ютерних об'єктів та процесів виконували в СКТБ математичних машин і систем, де розробляли малі ЕОМ, вони тривають і нині в однойменному інституті НАН України. Ці дослідження заклали основу комп'ютерної алгебри [6].

Графічне програмування. Академік В.М. Глушков підтримав розроблення графічної Р-метамови, яку вперше у практиці програмування було підтверджено авторським свідоцтвом. Її було реалізовано у вітчизняному пристрої синтаксичного контролю Р-програм і впроваджено в технологічному комплексі автоматизації програмування на машинах «БЭСМ» і ЄС ЕОМ [15], у якому І.В. Вельбицький під науковим керівництвом В.М. Глушкова запровадив апаратну реалізацію синтаксису і семантики мови програмування. Це визначило фактичне застосування Р-графічного методу програмування в оборонній промисловості СРСР.

Композиційне програмування, запропоноване В.Н. Редьком [18], орієнтоване на створення в рамках системи ДЕФІПС програми з функцій за ланцюжком: «дані — функція — ім'я функції — композиція — дескрипція». Тріада «дані — функція — композиція» задає семантичний аспект програми, а «дані — ім'я функції — дескрипція» — синтаксичний аспект. Ця система містить способи побудови функцій за заданими ланцюжками та множинами іменованих даних, дескрипцій і денотатів. Її застосовують для семантичного конструювання програм з використанням функцій декомпозиції, ординарних функцій оброблення операцій та інтерфейсних функцій для зв'язку арних функцій та іменованих даних. Система забезпечує зв'язок із сучасними СУБД для оброблення даних реляційного типу в композиційних програмах. Нині ці ідеї розвивають на факультеті кібернетики КНУ імені Тараса Шевченка в межах нової дисципліни — програмології.

Ідея індустрії програм. В.М. Глушков ініціював застосування алгебраїчних перетворень, доказу теорем, системного проектування рекурсивних, конвеєрних і макрокон-

веєрних машин для організації високоефективних обчислень та побудови АСУ з використанням готових програм, накопичених у фондах. Ідею індустріального принципу Віктор Михайлович сформулював на одному з наукових семінарів, що відбувся 5 березня 1975 р.: «*Мине 20–30 років, програми випускатимуть, як на збиральному конвеєрі фабрики Форда, з готових «деталей». Фабрики програм працюватимуть за принципом збирання в автомобільній промисловості*». Він передбачав, що в майбутньому програмні продукти випускатимуть на індустріальній основі, як на конвеєрі, а фабрику програм буде оснащено технологічними лініями побудови складних програм із простіших з автоматизованими засобами їх підтримки.

Засоби автоматизації технології програмування почали активно розвивати в Інституті кібернетики (ІК) АН УРСР. З 1965 р. цей напрям став стратегічним для відділу, де працює автор статті. У зв'язку з побудовою нового обчислювального комплексу «Днепр-2» у КБ математичних машин і систем ІК АН УРСР з ініціативи В.М. Глушкова було створено наукову групу для розроблення ОС і трансляторів з мови Автокод і Алгол-60. Водночас було відкрито фінансування робіт з конструювання комплексу і створення нових засобів автоматизації програмування. За рекомендаціями Віктора Михайловича було видано низку постанов і директив РМ СРСР та профільних державних комітетів, спрямованих на створення цих засобів обчислювальної техніки і технології виготовлення систем автоматизації програмування як продукції виробничо-технічного призначення. Паралельно із зазначеними українськими проектами в багатьох науково-дослідних інститутах СРСР також активно розробляли технічні і програмні засоби автоматизації програмування для «БЭСМ» і серії машин М-20.

Завдяки державній підтримці в СРСР було сформовано такі актуальні напрями, як створення комп'ютерних, системних і програмних технологій, а також засобів їх автоматизації для побудови індустріальними методами програмних і комп'ютерних систем.

ПЕРІОД ФОРМУВАННЯ У СВІТІ
ТЕХНОЛОГІЧНИХ НАПРЯМІВ (1965–1975)

Здавна всі нові теорії, що з'являлися в природничих науках, втілювалися в життя завдяки застосуванню технологій. У загальному сенсі термін «технологія» як сукупність систематизованих знань, методів, процесів та порядку їх використання під час виробництва продукту з потрібними властивостями, вживають у всіх сферах діяльності людства.

У галузі комп'ютеризації й інформатизації у світі сформувалися такі види технологій: комп'ютерна, системна, інформаційна і технологія програмування, або програмна інженерія (Software Engineering, SE). Термін SE виник одночасно з появою у нас технологій програмування, а офіційно вперше його було вжито на конференції НАТО в 1968 р. на означення самостійної дисципліни, сформованої міжнародним комітетом спеціалістів ACM і IEEE (www.swebok.com).

Структура комп'ютерної науки. Комп'ютерна наука (Computer Science) сформувалася у США і СРСР у зв'язку з побудовою ЕОМ та різних комп'ютерів. Вона складається з таких технологічних дисциплін.

Комп'ютерна технологія — дисципліна з теорії та принципів побудови комп'ютерів, фреймворків, мікропроцесорів, кластерів, суперкомп'ютерів, їх системного забезпечення (ОС, трансляторів, компіляторів тощо) для підтримки процесів оброблення даних. Її теоретичними основами є теорії Тюрінга, фон Неймана, автоматів, алгоритмів та кібернетики Глушкова [20, 21]. У ній використовують математику, логіку, теорію аналізу і систем, на її основі побудовано фреймворки, багатопроцесорні, рекурсивні, макроконвеєрні комп'ютери, різні пристрої, блоки, мікросхеми, карти тощо. Цю технологію, або комп'ютеробудування за термінологією Б.М. Малиновського, докладно описано в його монографіях, присвячених академіку В.М. Глушкову.

Сучасний рівень розвитку комп'ютерної технології в десятки разів вищий, ніж системної та програмної технологій. Нині у світі взято курс на побудову суперкомп'ютерів,

кластерів, Grid-систем для виконання глобальних проектів. Останніми роками в ІК НАН України створено енергоефективний суперкомп'ютер СКІТ-4 для проведення масштабних обчислень найскладніших задач у різних галузях народного господарства.

Системна технологія — це теорія, методи та принципи побудови інформаційних, комп'ютерних, автоматизованих систем, а також систем керування. Вона ґрунтується на технології комп'ютерних систем для моделювання різних комп'ютерних застосувань і нових засобів керування інформаційними системами (ОС, БД, СУБД та ін.). В її основу покладено теорію АСУ й інформатики Глушкова [5, 21], а також головні принципи логіки, математики, комп'ютерних наук і методи, які застосовують в економіці, фінансовій, банківській діяльності тощо. Сьогодні ми маємо значні досягнення у створенні загальних автоматизованих систем, що об'єднують діяльність різних міжнародних організацій.

Програмна інженерія (технологія програмування) — це система методів, способів і дисциплін з планування, розроблення, експлуатації та супроводу програмного забезпечення, призначених для його промислового виробництва. Вона охоплює всі аспекти створення програмного забезпечення від початку формулювання вимог до розроблення продукту і його використання, супроводження й остаточного списання. Її основою є теорії алгоритмів, програмування, обчислень і розподіленої комунікаційної мережевої технології [23–30], а масове виробництво програмних продуктів ґрунтується на теоріях менеджменту, планування, регулювання процесів і ресурсів, тестування, вимірювання результатів, оцінювання ризику та якості [25]. Ця технологія розвивається в напрямі побудови індустріальних фабричних методів конструювання розподілених програмних, інформаційних та бізнес-систем масового використання.

Інформаційні технології з 90-х років стали базисом комп'ютерної інфраструктури

сучасних корпорацій, підприємств і державних органів управління, що вирішують різні завдання з оброблення інформації, зокрема глобального типу. На розроблення інформаційних технологій, підготовку висококваліфікованих ІТ-спеціалістів, підтримку інформаційних ресурсів і доступу до них через систему Інтернет виділяють неймовірні ресурси. Цілі й завдання інформатики з побудови інформаційних систем і технологій академік В.М. Глушков сформулював у своїй останній у житті монографії [5]. Ці ідеї й сьогодні мають велике значення, на їх основі створено сучасну інформаційну технологію документообігу [25].

Інформаційні системи — це комп'ютерні системи оброблення в пошукових системах Інтернету різноманітної інформації з підприємницької та бізнес-діяльності, в тому числі бухгалтерського обліку, документообігу на всіх рівнях управлінської діяльності [1, 5, 25]. Вони є засобом керування й оброблення інформації для забезпечення продуктивності й ефективності роботи різних організацій. Нині інформаційні пошукові системи стали головним інструментом пошуку, відбору і накопичення різних інформаційних ресурсів для масового застосування.

Отже, до складу комп'ютерних наук входять: комп'ютерна технологія (Computer Engineering), системна технологія (System Engineering) і технологія програмування, або інженерія програмування (Software Engineering). Ці дисципліни, а також інформаційні системи і технології створюють простір інформатики (рис. 1), в якому технологія програмування посідає центральне місце, забезпечуючи всі складники цього простору відповідними теоріями, методами та засобами розроблення, розгортання і конфігурування програмних продуктів.

Сфера комп'ютерної технології займає всю нижню частину простору (рис. 1), що відповідає апаратному забезпеченню, а інформаційні системи — частину простору вгорі, на рівні окремих організацій. Програмна інженерія охоплює широкий діапазон, пов'язаний із систематичним розробленням

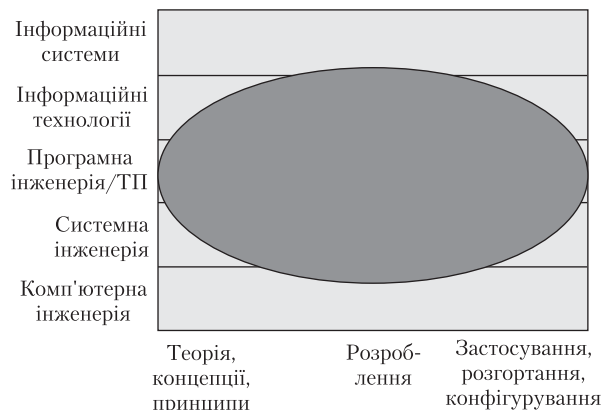


Рис. 1. Місце програмної технології в інформаційному просторі

програмного забезпечення, доменів, програмних проектів, моделей систем та методів їх створення. Сфера програмної технології поширюється вниз до системної технології, а вгору — до організаційних питань з проектування й розроблення інформаційних систем і технологій, що відповідають сучасним потребам різних організацій і підприємств.

Далі розглянемо лише зміст, засоби автоматизації технології програмування і динаміку її розвитку, починаючи з ідей академіка В.М. Глушкова.

ПЕРІОД СТАНОВЛЕННЯ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ (1975–1992)

На цьому етапі в ІК АН УРСР було зроблено вагомий внесок в індустрію створення програмних продуктів на великих ЕОМ. Розроблено методи, технології, інструментальні засоби, а саме: Р-технологію, збиральну технологію з модулів за інтерфейсами, технологію створення систем і пакетів прикладних програм.

Крім того, в СРСР було створено низку автоматизованих систем керування для потреб різних галузей промисловості, зокрема оборонної, які ґрунтувалися на теорії АСУ Глушкова. У 70-х роках на основі ідей В.М. Глушкова було побудовано перші АСУ (Львівський телевізійний завод, Сєвєродонецький хімкомбінат тощо), а також за його безпосередньої участі було створено АСУ



Стаття в журналі «Für Dich» про жінок-кібернетиків, зліва – К.М. Лаврішцева, 1972 р.

для металургійної промисловості з оброблення даних технологічних процесів на двох комбінатах Німеччини, у Берліні та Лейпцигу. Віктор Михайлович був членом Міждержавної комісії СРСР і НДР, а також Державної приймальної комісії АСУ ТП. За виконану роботу (1972) групу розробників з ІК АН УРСР, зокрема автора статті, було нагороджено орденом Дружби НДР за участь у створенні АСУ засобами обчислювального комплексу «Днепр-2».

Аналізуючи результати, досягнуті в цій галузі, В.М. Глушков сформулював три основні напрями її розвитку на майбутнє [4]:

- модульна система автоматизації програмування АПРОП зі стандартизованих програмних заготовок за принципом «знизу–вгору» і збирання їх у складні системи [2];
- метод формалізованих технічних завдань для проектування складних програмних комплексів з використанням кількох алгоритмічних мов для опису окремих блоків на рівнях послідовної деталізації проекту [1, 3–5];
- Р-технологія програмування для автоматизації проектування систем «згоризонт», починаючи з опису структур даних у системах керування [5, 15, 16].

Актуальним завданням Віктор Михайлович вважав розвиток технології комплексного проектування обчислювальних сис-

тем, коли проектування технічних засобів системи поєднане в один процес з проектуванням її базисного матзабезпечення. Цю ідею було реалізовано в системі ПРОЕКТ [3] і в серії ЕОМ «Мир» для інженерних розрахунків з використанням аналітичних перетворень. Перспективну тенденцію В.М. Глушков бачив і в переході від однопроцесорних фон-нейманівських машин до мозкоподібної машинної структури, до якої належала ЕОМ «Україна» [20, 21].

Перехід до виробництва систем. У цей період поширилася ідея повторного використання готових програм для розвитку на державному рівні індустрії програм. Було створено:

- республіканські й державні фонди алгоритмів і програм для обміну готовими напрацюваннями щодо реалізації різних завдань оброблення даних і подальшого їх розповсюдження за запитом зацікавлених користувачів;
- першу фабрику в м. Калінін (1978) з побудови програм для виробництва АСУ із готових програмних заготовок фондів алгоритмів і програм державного й республіканського значення;
- державні програми проведення НДДКР при ДКНТ і ДКОТІ СРСР з розроблення інструментальних систем автоматизації складних програмних комплексів і пакетів прикладних програм (АПРОП, ПРИЗ, ЯУЗА, СІГМАСТАТ, РТК-комплекс, Мультипроцесист, Макробол).

За постановами РМ СРСР, програми державних фондів алгоритмів і програм здобули статус виробів виробничо-технічного призначення. Фактично їх можна було використовувати за призначенням як самостійні програми для обчислення математичних і економічних задач. Проте повторне використання готових продуктів було утруднене через відсутність опису інтерфейсів і недоробки в алгоритмах. Завод у м. Калінін був першим дослідним виробництвом систем АСУ і працював упродовж кількох років. На ньому було побудовано два дослідні зразки АСУ — модернізовані

версії попереднього зразка. Розробити нові АСУ з використанням готових програм із фонду було складно через невелику кількість типових програм для АСУ, відсутність стандартизації та інтерфейсів.

Фонди алгоритмів і програм функціонували понад 20 років і припинили існування з розпадом СРСР. Цей період завершився вивезенням на звалище обчислювальної техніки (ЄС ЕОМ, прототипом якої є ІВМ, М-50, «БЭСМ» та ін.).

Концепція збирання модулів. Починаючи з 1975 р., в ІК АН УРСР почали формувати концепцію системи автоматизації виробництва програм (АПРОП) як шлях до індустрії програм [2, 8, 16, 26, 27]. Для розуміння сутності збирального конвеєра розробники як приклад розглядали стандарти в автомобільній промисловості. Головне, що було з'ясовано, — це наявність автоматизованих *ліній* конвеєра, які збирають окремі великі виробки з готових стандартних деталей, та *стикувальних* «болтів і гайок». В АПРОП «деталіями» були програмні модулі, які ще потрібно було стандартизувати, а для їх збирання винайти дещо, що замінювало б стикування і скріплення. Так виникла ідея проміжного *інтерфейсного перехідника* між двома модулями, який передає дані від одного модуля другому, отримує від нього результати і передає їх вихідному модулю. І якщо передані дані й результат за типом нееквівалентні вихідним даним, то перехідник трансформує їх у відповідну форму подання даних. Перехідник виявився гнучким зв'язком двох готових модулів, який не змінює змісту даних, а передає їх у потрібному вигляді, за потреби перетворюючи нееквівалентні типи.

Основою АПРОП були готові модулі, інтерфейси й методи збирання, які повністю відповідали ідеї *збирального конвеєра* Глушкова. Після смерті Віктора Михайловича розвиток збирального програмування підтримував академік АН СРСР А.П. Єршов.

Основні результати цього періоду полягають у розвитку напрямів автоматизації системної і програмної технології та формуванні нового виду програмування.

Розвиток інтерфейсу технології збирання. За ідеями В.М. Глушкова було сформовано новий вид програмування — збиральний, орієнтований на об'єднання різномовних модулів. Засобами їх об'єднання у великому фреймворку ЄС ЕОМ були: інтерфейс, модулі повторного використання, життєвий цикл технології збирання та збиральний конвеєр — автоматизована система збирання формалізованих модулів і програм у складніші програмні структури. Систему АПРОП розробляли понад 10 років за фінансової підтримки Міністерства радіопромисловості СРСР як частину технології ПРОТВА. Вона здобула Державну премію СРСР і була впроваджена в 52 організаціях у різних регіонах країни.

Головне нововведення збирального програмування — інтерфейс (міжмодульний, міжмовний і технологічний) [8, 30] та бібліотека інтерфейсних функцій перетворення типів даних у різних мовах і платформах. Перше у світі визначення поняття *інтерфейсу* та мови його опису було дано в 1976 р. науковою групою під керівництвом В.М. Глушкова під час проектування системи АПРОП [2]. Ідея інтерфейсу для зв'язку модулів значно випередила зарубіжні розробки — мова МІЛ (Module Interface Language) з'явилася там лише в 1983 р. Нині інтерфейс зберігає свою актуальність і виступає як головна домінанта взаємодіючих компонентів та об'єктів у сучасних глобальних і мережевих середовищах.

Міжмодульний інтерфейс — це посередник обміну даних між передавальним і приймальним модулями. *Міжмовний інтерфейс* — сукупність засобів і методів зв'язку структур і типів даних різних мов програмування. *Технологічний інтерфейс* — сукупність методів і засобів взаємозв'язку процесів і операцій технологічних ліній.

Методику створення технологічних ліній запропоновано нами в 1987 р. [26] і апробовано на 6 лініях автоматизованої інформаційної системи «Юпітер-470», що стало першою роботою з формалізації та застосування технологічних ліній у проектах розроблення

великих інформаційних систем. Подальшим розвитком цього напрямку є продуктивні лінії, запропоновані Інститутом програмної інженерії (США).

Розроблену нами концепцію ліній було автоматизовано за допомогою інтерфейсних модулів-посередників, які генерувала система АПРОП [8, 26], що сприяло скороченню обсягу робіт під час збирання різномовних програм через інструментарій генерації посередників. Набагато пізніше, в 1985–90-х роках, з'явилися зарубіжні мови опису інтерфейсів API, IDL, SIDL та ін. Їх використовують і дотепер у процесі створення нових програмних систем із готових програм, компонентів повторного використання і сервісів.

Академік АН СРСР А.П. Єршов свого часу зазначав, що збиральне програмування вирішує завдання багаторазового і швидкого застосування в процесі створення програми заздалегідь виготовлених «деталей». Для того щоб уявити корисність і важливість такого підходу, досить згадати про роль збірних конструкцій у житловому будівництві. Замість деталей у збиральному програмуванні використовують програмні модулі, що мають структуру й функціональну цілісність і разом з тим спеціально пристосовані до контрольованої інформаційно-логічної взаємодії (обміну інформацією) з іншими модулями. А.П. Єршов вважав, що збиральне програмування є ефективним, якщо комбінування порівняно невеликого числа заздалегідь запрограмованих модулів дає змогу швидко вирішити будь-яке завдання з певного класу проблем. Орієнтування на клас задач — особливість збирального програмування, що зумовлює його актуальність, оскільки широке розповсюдження міні- і мікро-ЕОМ дозволяє застосовувати кожен окрему машину для вирішення певних завдань.

Надалі збиральне програмування розвивалося шляхом індустрії готових програм, що цілком відповідає концепції конвеєрного збирання Глушкова на фабриках програм з технологічними лініями.

Результатами цього періоду розвитку технології програмування були:

- формування теорії і практики збирального програмування; побудова за збиральною технологією понад 500 прикладних програм оброблення даних у великому всесоюзному проекті «Юпітер-470» Міністерства радіопромисловості СРСР під керівництвом В.В. Ліпаєва та комплексу засобів автоматизованого проектування програм для різних ЕОМ, до складу якого входила система АПРОП [16, 26]. У 1987 р. за цей комплекс колективу авторів, у тому числі співробітникам ІК АН УРСР К.М. Лавріщевій і А.І. Нікітіну, було присуджено премію Ради Міністрів СРСР;

- формалізація технології модульно-інтерфейсного типу побудови пакетів прикладних програм під керівництвом академіка І.В. Сергієнка, орієнтованої на інтегровані, проблемні й інтелектуальні прикладні системи та їх сімейства, призначені для прийняття рішень у задачах оброблення даних на транспорті, в економіці, статистиці, математичному численні, системній оптимізації тощо. Побудовано і впроваджено понад 30 ППП, які працювали тривалий час [17].

ПЕРІОД РОЗВИТКУ ОБ'ЄКТНОЇ, КОМПОНЕНТНОЇ ТА АГЕНТНОЇ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ (1992–2002)

Цей період насамперед характеризувався появою і розвитком об'єктно-орієнтованого підходу (ООП) та мережі Інтернет. Після здобуття Україною незалежності на базі Інституту кібернетики було створено 5 інститутів, зокрема Інститут програмних систем (ІПС) НАН України, одним із ключових напрямів досліджень якого стали програмні технології. Перспективними напрямками розвитку технологій програмування було визначено такі:

- застосування об'єктно-орієнтованого підходу в проектуванні програмних систем;
- розроблення комп'ютерної технології проектування засобами CASE-систем (систем автоматизації з попереднього періоду);
- вивчення теорії набуття знань про об'єкти і функції у вигляді КПП;
- стандартизація методів інтеграції систем;

- розроблення нових підходів з подання інтерфейсу;
- інженерія якості та підходи щодо забезпечення надійності програмних систем.

Оцінюючи сьогодні зазначені напрями, можна стверджувати, що в Україні вони активно розвивалися в рамках фундаментальних проектів ІПС НАН України — від теорії й методології ООП до задання предметної області об'єктами і відношеннями між ними. Об'єкти-методи об'єднують у класи й підкласи, члени яких можуть «успадковувати» властивості інших об'єктів і бути поліморфними. Зв'язки між об'єктами задають за допомогою оператора віддаленого виклику RPC і мови опису інтерфейсного посередника. У ньому задають дані для передавання іншому об'єкту й операції звернення до готових методів застосувань на сервері. Апарат ООП — вищий рівень проектування, розроблення та інтеграції різних програмних об'єктів у розподіленому середовищі на зразок клієнт — сервер, що охоплює механізми реалізації та об'єднання різнорідних об'єктів у Sun Microsystems. Розроблено передову технологію побудови розподілених систем, об'єкти якої зберігаються на сервері, а звернення до них здійснюється засобами RPC і мовою опису інтерфейсів у клієнті [28].

Об'єктно-компонентне програмування. Об'єктний підхід Граді Буча (1989) приваблював усіх своєю математичною спрямованістю (об'єкти, наслідування, поліморфізм, класи, суперкласи тощо). Першу системну підтримку цього підходу було виконано в системі CORBA, потім у системах COM, DCOM, DSOM (1994). Основний засіб взаємодії різних об'єктів у системі CORBA — мова IDL, яку використовує брокер об'єктних запитів ORB CORBA для встановлення зв'язків між різнорідними об'єктами (C++, Java, Smalltalk, Cobol, VC++, Ada та ін.) у розподілених системах. Поряд з цим фірма «Rational Rose» створила десятки ООП-інструментів для широкого загалу. Об'єктність стала необхідним атрибутом нових мов програмування, відповідних систем програмування й бібліотек загальних систем (VS.Net, JAspect, JavaScript тощо) [23].

У рамках фундаментального проекту НАН України нами створено об'єктно-компонентний метод (ОКМ) [27, 28], в основу якого покладено нову теорію об'єктно-компонентного проектування моделей предметних областей з об'єктів, інтерфейсів і формальних правил перебудови об'єктної моделі в компонентну, визначену на відповідній множині компонентів, що реалізують методи об'єктів моделі, з яких збирають програмні системи. Компонентна модель доповнена формальними моделями взаємодії та варіабельності для забезпечення функціонування побудованої системи в сучасних гетерогенних середовищах на зразок Grid і Cloud Computing. Цей метод реалізовано в ІТК ІПС НАН України.

Інсерційне моделювання агентів і середовищ. Новий напрям у програмуванні, який є продовженням досліджень, започаткованих іще В.М. Глушковим, а нині його очолює учень і соратник Віктора Михайловича академік О.А. Летичевський. У цьому виді програмування керівний автомат відіграє роль агента, а інформаційний — середовища занурювання агента. Під агентами розуміють реальні фізичні, біологічні, соціальні та інші системи, під середовищем — набір дій агентів та множину їхніх станів.

Інженерія якості програмних систем. Наступним важливим досягненням і внеском в індустрію програмних продуктів стало розроблення основ інженерії якості програмного забезпечення за проектами ДКНТ. Було проведено дослідження міжнародного й вітчизняного досвіду з розроблення високоякісних програмних систем і відповідних стандартів. Важливу роль у цьому напрямі відіграв стандарт ISO/IEC 12207-96, в якому якість подано окремим процесом, що охоплює всі аспекти проектування, вимірювання й оцінювання показників якості продукту [24].

У результаті досліджень сформовано власний погляд на проблему якості й апробовано моделі в низці проектів Інституту, спільних з МЗС України, де особливо важливим є такий показник якості, як надійність.

У проектах ДКНТ за мету ставили розроблення стандартів якості. При КМ України створено організацію «Софт-рейтинг» для оцінювання і сертифікації програмних продуктів в Україні. Крім того, в ІПС НАН України під керівництвом академіка П.І. Антона розроблено 10 методик з якості програмного забезпечення, які було включено в замовлення МО України.

Отже, роботи цього періоду є значним внеском у розвиток сучасних технологій програмування в напрямі підвищення якості програмних продуктів і застосування індустріальних методів створення якісних програм масового використання.

ПЕРІОД РОЗВИТКУ ІНДУСТРІЇ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ (2002–2012)

Політика влади щодо індустрії програмних продуктів. Державне агентство з питань науки, інновацій та інформатизації, стурбоване фактичною відсутністю в Україні індустрії програмної продукції, у листопаді 2011 р. провело Міжнародний науковий конгрес з питань інфраструктури електронного уряду та ІТ-технологій. Було зазначено, що ця індустрія розвивається переважно завдяки діяльності закордонних фірм, яких в Україні налічується понад 1000. Силами вітчизняних фахівців вони виробляють програмні продукти для своїх країн. За такої політики розквіту цієї індустрії в Україні годі й очікувати. На нашу думку, потрібно укласти із закордонними компаніями договори на виготовлення продукції «50 на 50», тобто 50% вартості продукції, виробленої нашими фахівцями, слід залишати в нашій країні. Лише в такому разі можна розраховувати на прибутки. На жаль, Верховна Рада не приділяє уваги цьому питанню, хоча очевидно, що в державі має бути національна програма розвитку індустрії програмної продукції.

Розвиток індустрії програмної продукції. Цей період розвитку індустрії програм характеризується подальшим удосконаленням об'єктів збирання і ліній збирального конвеєра фабрик програм. З'явилися техно-

логії компонентного збирання [27, 28]; сформульовано принципи взаємодії компонентів у різних гетерогенних середовищах [29]; розроблено мультитехнології К. Чернецькі, лейтмотивом яких є перехід «від ручної праці до конвеєрного збирання»; створено різні фабрики програм. Аналіз сучасного стану цієї індустрії, проведений у роботі [30], свідчить про той факт, що нині у світі діють сотні фабрик програм та сервісів і спільними для всіх них є автоматизовані лінії збирання різних видів програмного забезпечення для масового використання.

Напрями розвитку фабрик програм. Фабрики програм почали розробляти з метою переходу до індустрії програмної продукції із заданими функціями, архітектурою, якістю та збільшення її продуктивності. Для цього було накопичено значну кількість різного роду готових програм та засобів з розроблення простих і складних продуктів. Основним механізмом фабрики є технологічні лінії, які відповідають стандартизованому життєвому циклу, причому на кожній лінії використовують свої засоби. Головна лінія розроблення складної продукції — це технологічна лінія збирання систем із готових програм, що знаходяться в різних бібліотеках і репозиторіях Інтернету.

Враховуючи досвід автоматизованого збирання різнорідних програм і аналіз сучасних зарубіжних фабрик індустріального типу, ми сформуваємо загальний набір елементів, що характеризують будь-яку фабрику програм:

- готові ресурси (артефакти, програми, сервіси, багаторазові компоненти тощо);
- специфікатор (інтерфейсний посередник) паспортних даних готових програмних ресурсів, описаний в одній із мов специфікації інтерфейсу;
- операційне середовище, що містить програмні засоби й інструменти для системного збирання різнорідних ресурсів;
- технологічні і продуктові лінії виробництва програмної продукції;
- метод розроблення і збирання компонентів повторного використання;

- збиральний конвеєр з набору ліній виробництва програм [30–34].

Сьогодні є всі необхідні умови для вирішення наукових і фізико-технічних завдань у рамках Європейського проекту Grid. У ньому функціонують фабрики програм системного і прикладного характеру, що відповідає базовій основі індустріального виробництва програмної продукції для глобальних наукових завдань [34, 35].

Комп'ютерні засоби для індустрії програмної продукції. Розвиток індустрії програм і систем зумовлений також створенням високоефективної елементної бази в багатоядерній процесорній конфігурації комп'ютерів, кластерів тощо. Оскільки комп'ютерна індустрія і в теоретичному, і в практичному аспекті значно випереджає розвиток індустрії програм, то індустріальна технологія забезпечить ефективне виконання масштабних обчислень складних задач у точних науках, АСУ і багатьох галузях промисловості. У зв'язку з цим з'явилися нові підходи й теорії, що характеризуються високим рівнем обчислювальної техніки і суперможливостями загальносистемного програмного забезпечення. Вони зумовили появу нової предметної мови опису систем (DSL, UML2, WorkFlow та ін.) зі сценаріями виконання й обчислення задач на даних із віртуальних міжнародних сховищ.

У роботі [34] наведено огляд сотень фабрик програм різного призначення й описано принципи їх побудови. Серед них відзначаємо фабрики, основані на конвеєрному збиранні, — фабрика К. Чернецькі, І. Бея, Дж. Грінфільда, Г. Ленца, М. Фаулера та студентська фабрика програм у КНУ імені Тараса Шевченка [33–36].

Важливим результатом цього періоду є формування теоретичних і практичних аспектів *компонентного програмування*. Розроблено теорію аналізу предметної області з об'єктів, створено з них об'єктну модель з використанням формальних математичних операцій визначення окремих елементів. Запропоновано метод трансформації об'єктної моделі в компонентну з переходом до ком-

понентів та інтерфейсів між ними [28]. Це дало можливість використати механізми збирання компонентів за граф-схемою модулів і тим самим узагальнити збиральне програмування. До операцій збирання входять зовнішня, внутрішня алгебри компонентів з формальними операціями на лінії побудови окремих компонентів. Об'єктно-компонентний метод розширено додатковою властивістю — варіабельністю для забезпечення заміни деяких компонентів новими функціями з метою отримання нового варіанта продукту [34–37]. Сформульовано модель варіабельності для сімейства програмних систем, яку реалізовано методом конфігураційного збирання в ІТК.

Технологічні лінії. Минув час, упродовж якого запропоновану вітчизняну методологію побудови технологічної лінії ми розвинули до розподілених середовищ і провели зіставлення з альтернативною закордонною методологією Product Lines SEI.

Методологію побудови лінії визначають на попередньому етапі технологічної підготовки робіт для створення спеціальної схеми лінії з процесів і операцій, що забезпечують продуктування елементів системи засобами мов програмування чи комплексу відповідних інструментів [28, 35]. Лінію комплектують з потрібних процесів життєвого циклу, що відповідають майбутній предметній області, стандартних інструментів, технологічних модулів і комплексу відповідного нормативно-методичного забезпечення. Для цього добирають готові прикладні ресурси і компоненти повторного використання, засоби й інструменти породження і реалізації окремих функцій або елементів програм (рис. 2). До процесів життєвого циклу відносять інженерію вимог, конструювання, тестування, оцінювання, керування якістю, проектом тощо. З них формують лінії для доменів.

Усі ресурси і процеси пов'язані технологічним маршрутом, який упорядковує і задає схему операцій і процесів, що підтримуються різними видами програмного забезпечення. На кінцевій операції маршруту виконують оцінювання якості продукту за

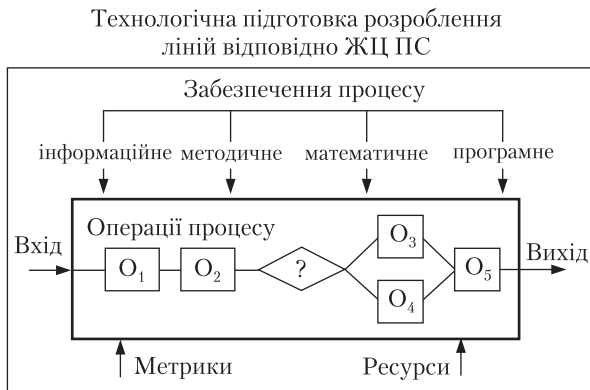


Рис. 2. Засади технологічної підготовки робіт з формування ліній

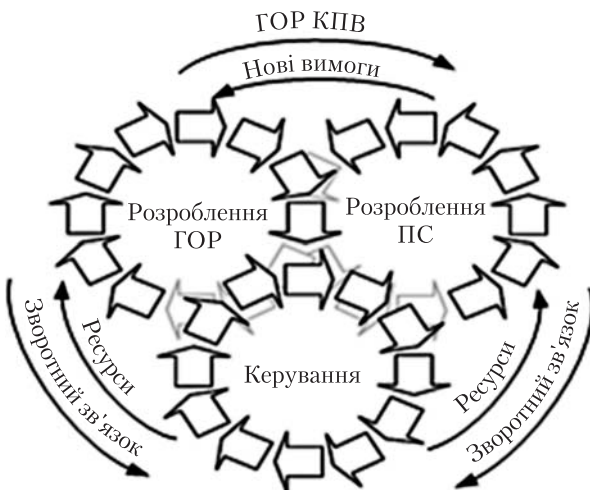


Рис. 3. Модель інженерного розроблення програмних систем

відповідною методикою [24]. Набір процесів створюють з урахуванням вимог міжнародного стандарту ISO/IEC 12207-96, 2007 та стандарту якості ДСТУ 3918-99. Процеси життєвого циклу, уніфікації тестування, оцінювання якості та керування проектами наведено на рис. 2. Методики описують спеціальною мовою з посиланнями на інструменти й технологічні модулі і правилами керування послідовною цілеспрямованою діяльністю спеціалістів з виконання процесів побудови елементів програми.

Альтернативна методологія — це методика побудови *продуктової лінії* (Product Lines)

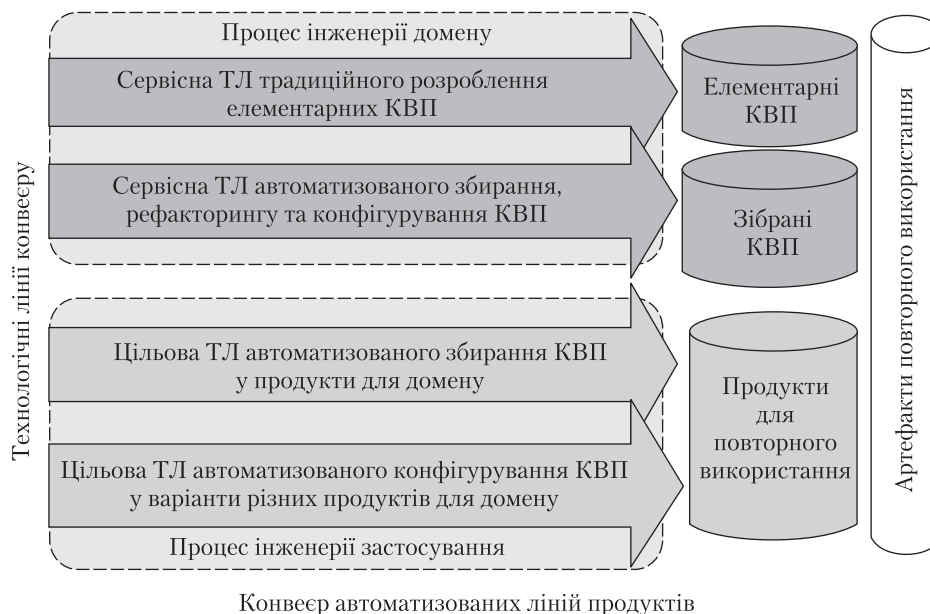
(www.sei.com/Productlines), що містить лінію продуктів і сімейство продуктів. Ці терміни визначено як групу продуктів або послуг, які мають спільну керовану множину властивостей, що задовольняють потреби певного сегмента ринку чи виду діяльності. Зазначена методологія ґрунтується на інженерній і процесній моделях, готових ресурсах, компонентах повторного використання.

Інженерна модель відповідає інженерії продукту за трьома видами діяльності — розроблення компонентів повторного використання чи готових ресурсів, збирання з них програмних систем і керування цими діями (рис. 3).

Розроблення готових ресурсів передбачає визначення сфери програмних систем, планування їх виробництва в контексті застосування, обмежень і стратегії виробництва. Збирання сімейства систем полягає в побудові плану реалізації кожної окремої системи на основі готових ресурсів та збирання їх за планом виробництва. Керування процесами ґрунтується на координуванні діяльності з випуску необхідних готових ресурсів і вирішенні завдань організаційного й технічного керування.

У *процесній моделі* виділяють множину процесів, виконуваних на двох рівнях — доменній інженерії, або інженерії програмної області, яку ще називають розробленням для забезпечення повторного використання (for reuse), та інженерії застосувань, або інженерії програмних систем, як розроблення з компонентів повторного використання (with reuse).

Сімейства програмних систем створюють з ліній збирання із застосуванням готових компонентів, і цей процес є кінцевою стадією в циклі підготовки процесів ліній, які відповідають вимогам і потребам домену, автоматизують процеси проектування компонентів з готових ресурсів (reuses, assets, artifacts тощо). Результат підготовки розроблення програмних систем — лінії конвеєра з підбору і збирання (агрегації) готових ресурсів повторного використання задля виробництва з них програмних продуктів для заданої предметної області (рис. 4).



Конвеєр автоматизованих ліній продуктів

Рис. 4. Структура конвеєра Глушкова

Кожна лінія збільшує продуктивність і поліпшує умови праці виконавців, скорочує час виконання робіт, підвищує якість продукції та рівень її готовності, знижує собівартість випуску. Саме про такий конвеєр мріяв академік В.М. Глушков і саме його ідею ми реалізували на студентській фабриці програм КНУ.

ПЕРІОД НАВЧАННЯ ТЕХНОЛОГІЯМ ПРОГРАМУВАННЯ (2000–2012)

Навчання технологіям програмування.

В.М. Глушков приділяв значну увагу навчанню кібернетичним наукам співробітників і студентів. Він читав лекції в Київському і Московському університетах, МФТІ та інших ВНЗ, написав велику кількість монографій, залишив нащадкам багато наукових праць з інформатики, які актуальні й сьогодні. У 1969 р. Віктор Михайлович створив у Київському університеті факультет кібернетики, заснував кафедри з теорії кібернетики, обчислень, програмування тощо, які очолювали його учні В.Н. Редько, А.В. Анісімов, О.К. Закусило та інші.

Відразу після появи Алголу-60 В.М. Глушков першим прочитав лекцію «напіванглійською» мовою для співробітників ІК АН УРСР

і студентів КНУ. Згодом К.Л. Ющенко розробила нормативний курс із Алголу-60 і доручила мені читати його студентам. Так у 1963 р. розпочалася моя педагогічна діяльність, яка триває й до сьогодні. Потім були курси з інших мов програмування, а з 80-х років – курс із технології програмування. Вивчаючи цю проблему з наукових позицій в ІК та ІПС НАН України, я постійно обговорюю зі студентами всі нові наукові питання, проводжу практичні лабораторні роботи з сучасних інструментальних систем і засобів, залучаючи молодь до створення нових розробок.

У 2007 р. Кабінет Міністрів України відповідно до міжнародної програми Computing Curricula затвердив навчальний курс «Програма інженерія», який за змістом близький до курсу з технології програмування. Обидва ці нормативні й прогресивні курси я викладаю на факультеті кібернетики КНУ, на кафедрах «Інформаційні системи» та «Теорія і технологія програмування». До них входять лекційні й лабораторні заняття з теорії і практики розроблення програм та артефактів сучасними мовами програмування для різних середовищ, вивчення сучасних загальних ОС.



Студентську фабрику програм демонстрували на Міжнародному науковому конгресі. Зліва направо: А. Дзюбенко, К.М. Лавріщева, А. Аронов, 2011 р.

Студенти захищають магістерські роботи з сучасних питань визначення моделей якості компонентів повторного використання, взаємодії систем і середовищ, варіабельності й живучості систем, обчислення задач для Grid і Cloud Computing, моделей даних за стандартом ISO/IEC GDT 11404-2007, які накопичують у віртуальних і глобальних сховищах даних для потреб індустрії програмних систем.

Студентська фабрика програм КНУ. За концепцією збирального конвеєра В.М. Глушкова в 2011 р. вперше в Україні за участю студентів було побудовано фабрику програм і артефактів [35–37]. Вона орієнтована на вивчення всіх аспектів програмної інженерії, побудови інформаційних і програмних систем з використанням розроблених студентами артефактів і програм за тематикою предметних лабораторних і дипломних робіт.

Фабрику програм створено силами студентів у процесі вивчення дисциплін «Технологія програмування інформаційних систем» і «Програмна інженерія» на 4-му курсі факультету кібернетики КНУ під керівництвом автора [31–37]. Вона доступна в Інтернеті з 2011 р. (<http://programsfactory.univ.kiev.ua>). Основу фабрики становлять готові програмні продукти, що специфікуються в мові WSDL, зберігаються в репозиторії і можуть поповнюватися за допомогою готових бібліотек на зразок Matlab, Demral тощо. Фабрика обладнана такими лініями:

- програмування в C# VS.Net з отриманням готового продукту і збереженням його в стандартному вигляді в репозиторії;
- добір готових артефактів і програм з репозиторію для подальшого використання;
- збирання готових ресурсів у складні структури;
- навчання методології проектування систем і сімейств систем за е-підручниками «Програмна інженерія» КНУ та на сайті www.intuit.ru.

До 90-річчя з дня народження академіка В.М. Глушкова заплановано додати низку новітніх ліній:

- генерація прикладних систем за новою предметною мовою DSL із застосуванням інструментів Eclipse–DSL, Tool DSL Microsofts та ін.;
- трансформація загальних типів даних GDT у фундаментальні FDT на основі стандартних засобів генерації даних стандарту ISO/IEC 11404-2007 для використання в різномовних розподілених системах;
- нові навчальні дисципліни (домен програмної інженерії, життєвий цикл та ін.);
- побудова розподілених систем із сервісів і веб-сервісів;
- підключення нових продуктових ліній за механізмами Product Lines SEI, виготовлених студентами.

Фахівці різних спеціальностей мають змогу обирати для потреб своєї наукової діяльності необхідні програми й компоненти. Тому особливо важливо мати такі фабрики при університетах: з одного боку, ними мо-

жуть користуватися студенти інших вишів, а з другого — молодь бере участь у розвитку індустрії програмних продуктів.

Комплекс ІТК [38]. Запропонований теоретичний апарат компонентного програмування реалізовано за участю студентів МФТІ в ІТК (www.sestudy.edu-ua.net), до складу якого включено реалізовані моделі взаємодії через плагіни Eclipse, а саме:

- *VS.Net, Eclipse* для реалізації взаємодії окремих програм між собою;
- *Corba, MS.Net* для реалізації взаємодії програм із цих середовищ між собою;
- *IBM VSphere, Eclipse* для взаємодії програм між собою в цих середовищах.

Головне призначення цього ІТК — опис і подання понад 10 спрощених технологій виробництва програмних систем із готових компонентів, а також ліній навчання нормативному курсу з технології програмування за допомогою електронного підручника. Крім того, на сайті ІТК розміщено для ознайомлення і навчання наукові статті з фундаментальних основ індустрії програмних продуктів, е-монографію [28] фахівців відділу «Програмна інженерія» ІПС НАН України щодо теоретичних і прикладних аспектів виготовлення систем з готових ресурсів, накопичених у репозиторії, а також запропонованих методів індустрії програмних продуктів, що ґрунтуються на технологічних лініях.

ВИСНОВКИ

Розглянуто теоретичні й прикладні аспекти технології програмування, започаткованої академіком В.М. Глушковым, які набули розвитку в нових теоріях, технічних і програмних інструментах. Фабрична індустрія програмних продуктів, затребувана сьогодні інформаційною спільнотою, фактично стала реалізацією концепції збирального конвеєра В.М. Глушкова. Для її втілення було розроблено методологію побудови збирального конвеєра для індустріального виготовлення різних типів програмної продукції, яка свого часу значно випередила появу нової ідеології, а саме продуктової лінії, в Інституті програмної інженерії США (2004).

Важливим кроком у цьому напрямі є розроблені в ІПС НАН України і КНУ імені Тараса Шевченка сайти фабрики програм та ІТК, призначені для студентів і розробників різного роду програм в Україні й СНД, подібно до використання промислових фабрик програм.

Перспективними напрямками розвитку технології програмування є побудова нових ліній для створення розподілених прикладних систем з веб-сервісів і семантик у середовищі веб-семантик; розроблення методології автоматизованої побудови гнучких збиральних технологій типу «нано» (тобто технології виробництва елемента із заданою «атомарною» структурою, якою можуть бути маленькі компоненти чи сервіси) для реалізації глобальних задач та їх обчислень у сучасних середовищах на зразок Grid і Cloud Computing.

СПИСОК ЛІТЕРАТУРИ

1. *Капитонова Ю.В., Летичевский А.А.* Парадигмы и идеи академика В.М. Глушкова. — К.: Наук. думка, 2003. — 456 с.
2. *Глушков В.М., Лаврищева Е.М., Стогний А.А. и др.* Система автоматизации производства программ (АПРОП). — К.: ИК АН УССР, 1976. — 134 с.
3. *Глушков В.М., Капитонова Ю.В., Летичевский А.А.* О применении метода формализованных технических заданий к проектированию программ обработки структур данных // Программирование. — 1978. — № 6. — С. 5–12.
4. *Глушков В.М.* Фундаментальные основы и технология программирования // Программирование. — 1980. — № 2. — С. 13–24.
5. *Глушков В.М.* Основы безбумажной информатики. — М.: Наука, 1982. — 552 с.
6. Системы компьютерной алгебры семейства АНАЛИТИК. Теория. Реализация. Применение. — К., 2010. — 762 с.
7. *Ющенко Е.Л.* Адресное программирование. — К.: Техн. л-ра, 1963. — 286 с.
8. *Лаврищева Е.М., Борисенко Л.Г., Гришкевич К.И. и др.* Транслятор с языка Д-АЛГАМС для УВК «Днепр-2». — К.: ИК АН УССР, 1970. — 186 с.
9. *Каминин С.С., Любимский Э.З.* Алгоримический машино-независимый язык АЛМО для задач символьной обработки // Тр. I Всесоюз. конф. по программированию. — К., 1968.
10. *Иванников В.П.* Язык системного программирования. — М.: ИТМ, ОТ, 1959.
11. Информационные материалы и тезисы пленарных докладов II Всесоюз. конф. «Технология программирования». — К.: ИК АН УССР, 1986.

12. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. — К.: Наук. думка, 1974. — 318 с.
13. Ершов А.П. Введение в теоретическое программирование. — М.: Наука, 1977. — 288 с.
14. Тыгу Э.Х. Концептуальное программирование. — М.: Наука, 1984. — 256 с.
15. Вельбицкий И.В., Ходаковский В.Н., Шолмов Л.И. Технологический комплекс автоматизации программ на машинах ЕС ЭВМ и БЭСМ-6. — М.: Статистика, 1980. — 263 с.
16. Лаврищева Е.М., Грищенко В.Н. Связь разноязыковых модулей в ОС ЕС. — М.: Финансы и статистика, 1982. — 127 с.
17. Редько В.Н., Сергиенко И.В., Стукало А.И. Пакеты прикладных программ. — К.: Наук. думка, 1992. — 317 с.
18. Редько В.Н. Композиции программ и композиционное программирование // Программирование. — 1978. — № 5. — С. 17–26.
19. Молчанов И.Н. Машинные методы решения задач прикладной математики. — К.: Наук. думка, 1987. — 255 с.
20. Глушков В.М. Теория алгоритмов. — К.: КВИРТУ, 1961. — 167 с.
21. Глушков В.М. Кибернетика, вычислительная техника, информатика. — К.: Наук. думка, 1990. — С. 262–281.
22. Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А. Алгеброалгоритмические модели и методы параллельного программирования. — К.: Академперіодика, 2007. — 634 с.
23. Андон Ф.И., Лаврищева Е.М. Методы инженерии распределенных компьютерных приложений. — К.: Наук. думка, 1997. — 229 с.
24. Андон Ф.И., Коваль Г.И., Коротун Т.М. и др. Основы инженерии качества программных систем. — К.: Академперіодика, 2007. — 680 с.
25. Задорожна Н.Т., Лаврищева К.М. Менеджмент документообігу в інформаційних системах освіти. — К.: Педагог. думка, 2007. — 220 с.
26. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика. — К.: Наук. думка, 2006. — 454 с.
27. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Основы индустрии программных продуктов. — К.: Наук. думка, 2009. — 372 с.
28. Лаврищева К.М., Коваль Г.И., Бабенко Л.П. та ін. Нові теоретичні засади технології виробництва сімейств програмних систем у контексті генерувального програмування. — К.: ІПС НАНУ, 2011. — 377 с.
29. Андон П.І., Лаврищева К.М. Розвиток фабрик програм в інформаційному світі // Вісн. НАН України. — 2010. — № 10. — С. 15–41.
30. Бабенко Л.П., Лаврищева К.М. Основы програмної інженерії. — К.: Знання, 2001. — 269 с.
31. Лаврищева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения. — М.: МОН РФ, 2007. — 415 с.
32. Лаврищева К.М. Програмна інженерія. — К.: Академперіодика, 2008. — 319 с.
33. Lavrisheva E., Ostrovski A. General Disciplines and Tools for E-Learning Software Engineering. — <http://senldogo0039.springer-sbm.com/ocs/>.
34. Аронов А.О., Дзюбенко А.І. Підхід до створення студентської фабрики програм // Проблеми програмування. — 2011. — № 3. — С. 42–49.
35. Лаврищева К.М., Слабостицька О.О., Коваль Г.І., Колесник А.Л. Теоретичні аспекти керування варіабельністю в сімействах програмних систем // Вісн. КНУ. — 2011. — № 1. — С.151–158.
36. Андон П.І., Лаврищева К.М. Методологія побудови ліній виробництва програмних продуктів і їх застосування // Інформаційне суспільство в Україні: матер. міжнар. наук. конгр. (25–26 жовтня 2012, Київ, Україна). — С. 19–26.
37. Лаврищева К.М. Інструментально-технологічний комплекс для розробки й навчання прийомам виробництва програмних систем // Вісн. НАН України. — 2012. — № 3. — С. 17–26.
38. Лаврищева К.М., Зінкович В.М., Колесник А.Л. та ін. Інструментально-технологічний комплекс для розробки й навчання прийомам виробництва програмних систем. — Свідоцтво про реєстрацію авторського права на твір № 45292 від 27.08.2012.

Стаття надійшла 10.01.2013 р.

Е.М. Лаврищева

Институт программных систем
Национальной академии наук Украины
просп. Академика Глушкова, 40, Киев, 03187, Украина
РАЗВИТИЕ ИДЕЙ АКАДЕМИКА В.М. ГЛУШКОВА
ОТНОСИТЕЛЬНО ТЕХНОЛОГИИ
ПРОГРАММИРОВАНИЯ

Статья посвящена 90-летию со дня рождения академика НАН Украины и АН СССР Виктора Михайловича Глушкова, его вкладу в развитие информационных технологий и технологии программирования с линиями производства программных продуктов из готовых ресурсов. Еще в 70-е годы прошлого века В.М. Глушков считал, что вскоре появятся фабрики программ, которые будут работать по принципу сборки, как в автомобильной промышленности, и со временем его предвидения полностью оправдались. В статье в исторической последовательности рассмотрены этапы развития технологии программирования, описана созданная на основе концепции В.М. Глушкова первая студенческая фабрика программ, очерчены перспективы дальнейшего развития этого направления.

Ключевые слова: компьютерная технология, системная технология, информационная технология, технология программирования, программная инженерия, информатика, сборочный конвейер, методология сборки.

K.M. Lavrisheva

Institute of Program Systems
of National Academy of Sciences of Ukraine
40 Glushkov Ave., Kyiv, 03187, Ukraine

DEVELOPMENT OF V.M. GLUSHKOV'S IDEAS
IN THE FIELD OF PROGRAMMING
TECHNOLOGY

The article is devoted to the 90th anniversary of NAS and AS USSR academician Victor Glushkov, his contribution to the development of information technologies and programming technology with lines of production of software products from the ready-made resources. As far

back as in the 70th years of past century Glushkov considered that the program factories working on principle of assembling, as in the motor industry, will appear before long, and with the lapse of time his foresights were fully justified. In the article in historical sequence the stages of development of programming technology are considered, the first program factory created on Glushkov's principles is described and perspective ways of development of this direction are outlined.

Keywords: computer technology, system technology, information technology, programming technology, software engineering, computer science, assembly line, assembly methodology.