

Незважаючи на те, що запропонована інтелектуальна інформаційна система має обмежений характер, при певному адаптуванні вона може бути застосована як оболонка для реалізації в умовах реальної страхової компанії.

1. Берко А.Ю. Інформаційні моделі прийняття рішень в медичному страхуванні // Вісник Національного університету "Львівська політехніка". – 2002. – № 464. – С. 3–11. 2. Берко А.Ю., Зведенюк С.І. Інтелектуальні технології в Інтернет-страхуванні // Вісник Національного університету "Львівська політехніка". – 2003. – № 489. – С. 3–12. 3. Пономаренко В.С. Інформаційні системи і технології в економіці. – К.: Академія, 2002. 4. Информационные технологии в страховании <http://www.rbc.ru/insurance/einsurance.html>.

УДК 681.3.06

Є.В. Буров

Національний університет "Львівська політехніка",  
кафедра інформаційних систем та мереж

## СИСТЕМА ФОРМАЛЬНИХ СПЕЦИФІКАЦІЙ ДЛЯ КОНФІГУРУВАННЯ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

© Буров Є.В., 2004

**Запропоновано формальні специфікації для автоматизації проектування та конфігурування розподіленої інформаційної системи.**

**This paper proposes formal specifications for automatic design and configuration of complex corporative information system.**

### Постановка проблеми у загальному вигляді

Розвиток інформаційних технологій сучасності сприяє збільшенню ролі інформаційних систем у виробничій діяльності будь-якої організації чи підприємства. У багатьох випадках від якості функціонування інформаційної системи (ІС) безпосередньо залежить якість виконання головних бізнес-процесів. Водночас ріст складності таких систем, що нерідко містять тисячі компонент різних виробників, відсутність засобів динамічного контролю за процесом функціонування роблять актуальною задачу побудови автоматизованих систем проектування та керування в інформаційних комп'ютерних мережах.

Важливою тенденцією у розвитку систем проектування та керування для інформаційних комп'ютерних мереж є перехід на новий рівень їх аналізу – рівень бізнес-процесів. Параметри якості виконання бізнес-процесів в першу чергу цікавлять будь-якого замовника та безпосередньо визначають прибутковість підприємства та успішність функціонування організації. Поняття бізнес-процесів покладено в основу головних стандартів що визначають рівень якості роботи підприємства – ISO 9000, CMM та нового стандарту ISO/IEC 15504.

Аналіз, проектування та керування інформаційною системою на рівні бізнес-процесів дозволяють визначити вплив якості проектних, структурних, функціональних вирішень та поточних налаштувань інформаційної системи як однієї з допоміжних підсистем підприємства на якість виконання головних бізнес-процесів.

Одним з шляхів вирішення вказаних проблем є побудова автоматизованих систем проектування та керування інформаційними системами. Спираючись на комплекс формальних моделей, широко застосовуючи засоби штучного інтелекту, експертні системи дозволять покращити якість проектування ІС, гнучкіше та оперативніше реагувати на зміни.

## Аналіз останніх досліджень

У роботі [1] запропоновано систему формальних специфікацій для автоматизованого проектування та моделювання розподілених інформаційних систем. ІС в [1] описана трьома специфікаційними мережами, між якими встановлено відношення:

$$S = \{NPc, NPr, NPt, NDv\},$$

де *NPc* – мережа процесів, *NPr* - мережа процесорів, *NPt* - мережа прототипів, *NDv* - мережа компонент. Процес проектування починають з визначення структури бізнес-процесів підприємства та виокремлення процесів, які відбуваються за участі інформаційної системи. Будують параметричні моделі окремих складових операцій мережі процесів.

На ранніх етапах проектування під процесом розуміють бізнес-процес, певну процедуру, зрозумілу Замовнику. На подальших етапах проектування процес – це операція, яка супроводжується перетворенням інформації (матеріальних потоків, параметрів системи). Мережа процесів визначає послідовність операцій у системі, їх взаємозв'язок.

Мережа процесорів *NPr* є подальшим кроком у побудові специфікації системи. Специфікація ускладнюється за рахунок розкриття структури розподілених процесів, визначення та ідентифікації компонент багаторазового використання – системних сервісів, а також визначення та опису об'єктів-виконавців процесів (процесорів). На цьому етапі приймають загальноархітектурні рішення.

На етапі побудови мережі прототипів *NPt* кожному процесору ставиться у відповідність певний прототип пристрою, програмного забезпечення або технічного рішення. При цьому використовують прості експертні системи.

При побудові мережі компонент *NDv* для кожного прототипу визначають марки та моделі реальних пристроїв або компонент програмного забезпечення, виконують необхідні вимірювання та експерименти.

Після побудови усіх специфікаційних мереж стає можливими проектування окремих підсистем та вирішення прикладних задач з налаштування ІС.

### **Виділення не вирішених раніше частин загальної проблеми, котрим присвячується означена стаття**

Мережа компонент – це остання специфікаційна мережа. Вона будується на базі закінченої специфікації мережі прототипів. Об'єктами, що розглядаються на цьому етапі, є апаратні компоненти (пристрої), програмні компоненти, персонал та інші об'єкти, задіяні у виконанні функцій системи. Для специфікацій цього етапу характерна найбільша деталізація моделей, орієнтація на реальні параметри керування та налаштування, використання моделей реальних пристроїв. Загалом етап можна поділити на дві частини, які розрізняються метою та набором задач, що вирішуються. Це етап побудови мережі компонент та етап експлуатації системи.

### **Цілі статті**

Розглянемо задачі етапу побудови мережі компонент. Вхідними даними для них є мережа прототипів – множина визначених рішень на рівні архітектур та типів компонентних рішень. Задача побудови мережі полягає у виборі для кожного типового рішення компонента з каталогу. Вибір відбувається на основі інформації про тип прототипу та тип компонента. Вирішення такої задачі на практиці ускладнюється такими обставинами:

– крім відповідності за типами, між прототипом та компонентом має бути і параметрична відповідність. Вимоги до об'єкта, виражені в параметрах прототипу, мають відповідати технічним параметрам об'єкта;

– в процесі вибору враховуються загальні (формальні та неформальні) правила вибору. Такими правилами, наприклад, може бути дотримання принципу придбання обладнання одного виробника, надання переваг продуктам певних виробників та ін.;

- вибране обладнання може недостатньо ефективно взаємодіяти з іншим обладнанням або взагалі бути несумісним з ним;
- вибрана конфігурація обладнання може змінити ступінь привабливості окремих компонент – кандидатів в наступних операціях вибору. Тому значення має і порядок розгляду прототипів. Рекомендується спочатку розглядати контейнерні об'єкти, а потім – об'єкти, що входять до них. Щодо об'єктів одного рівня, то вони розглядаються згідно з деяким порядком розгляду типів об'єктів, визначеним заздалегідь.

Процес вибору проводять, заносючи об'єкти-кандидати у тимчасову таблицю, в якій для кожного об'єкта визначено нормований показник привабливості. На підставі цих показників система пропонує варіанти вибору для проектувальника. Остаточний вибір робить проектувальник. Процес вибору має ітераційний характер. Можливе повернення на попередні стадії вибору, зміна рішення на них.

На етапі побудови мережі компонентійних об'єктів працюють з такими специфікаційними об'єктами:

- мережа пристроїв. Це результат вирішення задачі. Мережа складається зі специфікацій:
  - пристроїв;
  - зв'язків між ними;
  - каталогу пристроїв;
- таблиці правил вибору. Реалізують загальні правила вибору (наприклад, орієнтація на продукти одного виробника та специфічні правила вибору, коли вибір однієї компоненти змінює пріоритет вибору інших);
  - таблиці сумісності;
  - таблиці правил параметричних відображень. Вирішують задачу відображення параметрів прототипу у параметри конкретного пристрою.

Процес побудови мережі компонент полягає в послідовному перегляді всіх прототипів згідно з їх вкладеністю та порядком типів. Для кожного такого прототипу з каталогу компонент вибирають всі відповідні за типом об'єкти. Для кожного такого об'єкта:

перевіряється його параметрична відповідність та придатність. Для цього використовують таблиці правил параметричних відображень. Якщо потрібні параметри виходять за межі допустимого, варіант відкидають;

варіант перевіряють на відповідність правилам з таблиці правил. У випадку невідповідності – варіант відкидають (або для необов'язкових правил – залишають вибір за проектувальником);

перевіряють сумісність з елементами оточення (контейнерним елементом та взаємодіючими з об'єктом-кандидатом елементами). У випадку несумісності – варіант відкидають;

варіант вважають обраним.

Функції проектування рівня компонент подані сьогодні в декількох продуктах. Так, наприклад, в одному з функціонально найповніших продуктів (*Netcrack*) побудовано схему комп'ютерної мережі з орієнтацією на реальні пристрої. Продукт має велику бібліотеку описів таких пристроїв. При побудові сполучень він перевіряє пристрої на сумісність за їх інтерфейсами. Для готових схем реалізовані прості програми моделювання інформаційних потоків з оцінкою ступеня завантаженості пристроїв. Водночас продукт не дозволяє змінювати та модифікувати моделі, вносити нові типи моделей, сфера його застосування обмежена комп'ютерними мережами, він не враховує неформальні переваги та досвід проектування подібних систем, він не орієнтується на бізнес-процеси, не дозволяє вирішувати задачі аналізу та керування системами.

### **Основний матеріал**

Розглянемо детальніше дані, за якими складаються задачі побудови мережі об'єктів компонентії.

Мережа об'єктів компонентії NDv – це база даних про пристрої та програми, інші об'єкти, які використовуються у системі, їх експлуатаційні параметри та параметри налаштування, взаємозв'язки та відношення. Після побудови мережі компонент вона використовується для вирішення задач

керування та моделювання. Мережу компонент можна подати як сукупність описів об'єктів та їх зв'язків:

$$NDv = \{M(Dv), M(LDv)\}.$$

Кожен компонент  $Dv$  є екземпляром певного об'єкта-шаблону з каталогу компонент. Він може мати власну модель функціонування, скопійовану з шаблону, доповнену своїми параметрами, або ж тільки посилання на модель шаблону.

$$Dv = \{idDv, idPt, idTmDv, DtDv, M(InDv), PbDv, [MdDv], CmDv\},$$

де  $idDv$  – ідентифікатор об'єкта,  $idPt$  – ідентифікатор прототипу, якому відповідає об'єкт. Між прототипами та об'єктами є взаємно однозначне відношення.  $idTmDv$  – ідентифікатор відповідного шаблону,  $DtDv$  – структура об'єкта,  $M(InDv)$  – множина інтерфейсів об'єкта,  $PbDv$  – блок параметрів,  $MdDv$  – модель, або  $idMd_sDv$  – ідентифікатор моделі шаблону об'єкта.  $CmDv$  – коментар.

Ідентифікатор шаблону  $idTmDv$  пов'язує об'єкт з відповідним шаблоном. Структура об'єкта компоненті описує об'єкти та їхні взаємозв'язки, що входять у даний об'єкт.

$$DtDv = NDv' \subset NDv.$$

Деталізація об'єкта є підмережею об'єктів, для яких даний об'єкт є контейнером.

Інтерфейси об'єкта  $InDv$  виділяються окремо, так як вони визначають взаємодію об'єкта з іншими об'єктами та характеристики цієї взаємодії. Параметри інтерфейсів повинні відповідати параметрам зв'язків прототипа.

$$InDv = \{idInDv, TyInDv, PbInDv, CmInDv\},$$

де  $idInDv$  – ідентифікатор інтерфейсу,  $TyInDv$  – тип інтерфейсу,  $PbInDv$  – блок параметрів інтерфейсу,  $CmInDv$  – коментар. Інтерфейси об'єкта визначають під час вирішення задачі вибору об'єкта. При цьому враховують:

- вимоги прототипу (типи та характеристики вхідних та вихідних потоків та зв'язків);
- можливості шаблону об'єкта (можливі типи інтерфейсів, якими конфігурується даний об'єкт);
- правила параметричних відображень (вони дають змогу відобразити параметри прототипу в параметри об'єкта);
- таблиці сумісності (визначають сумісність та бажаність кожного наступного вибору залежно від результатів попереднього вибору).

Блок параметрів  $PbDv$  має типову для таких блоків структуру, тобто складається з параметрів, методів роботи з ними, параметричних залежностей.

Розрізняють:

- параметри налаштування,
- експлуатаційні параметри.

Існують відповідності між параметрами об'єктів-компонент та параметрами специфікаційних об'єктів інших рівнів. Ці залежності існують у формі параметричних залежностей для відповідних параметрів.

Об'єкти-компоненти не існують незалежно один від одного. Вони пов'язані не тільки інформаційними потоками, зв'язками, характерними для вищих рівнів специфікації, але між ними існують відношення між і на рівні об'єктів компонент. Ці відношення є носіями параметрів. Наприклад, між об'єктами 'комп'ютер' та 'мережевий адаптер' є відношення вкладеності (plug-in), а між комп'ютером та модемом, приєднаним до послідовного порта – відношення під'єднання.

Зв'язок описується набором параметрів.

$$LDv = \{idDv_1, idDv_2, TyLDv, PbLDv, CmLDv\},$$

де  $idDv_1$ ,  $idDv_2$  – ідентифікатори з'єднаних об'єктів,  $PbLDv$  – блок параметрів з'єднання,  $CmLDv$  – коментар.

Параметр типу сполучення TuLDv визначає типову конфігурацію параметрів сполучення з врахуванням типів обох з'єднаних об'єктів.

Каталог шаблонів об'єктів компонентії – це база даних, що містить описи шаблонів. Інформація за кожним шаблоном подібна до інформації про відповідний об'єкт, але має й такі відмінності:

– шаблон містить відповідності одному або декільком типам прототипів, тобто працює не з конкретними прототипами, а з об'єктами з каталогу прототипів;

– шаблон може містити цілу множину варіантів конфігурації (наприклад, додаткових блоків з різними інтерфейсами), тобто являти собою цілу групу можливих об'єктів;

– шаблон містить також і параметри, задані у вигляді інтервалів вибору, тоді як відповідний об'єкт-компонент – конкретне значення.

Каталог шаблонів об'єктів-компонент – це множина описів:

$$LiTmDv = M(TmDv).$$

Шаблон об'єкта – це кортеж

$$TmDv = \{idTmDv, M(TsTmDv), CmTmDv\},$$

де  $idTmDv$  – ідентифікатор шаблону,  $M(TsTmDv)$  – множина можливих конфігурацій,  $CmTmDv$  – коментар.

Кожна конфігурація  $TsTmDv$

$$TsTmDv = \{idTsTmDv, CfTsTmDv, InTsTmDv, PbTsTmDv, CmTsTmDv\}$$

містить ідентифікатор  $idTsTmDv$ , опис конфігурації  $CfTsTmDv$ , інтерфейсну частину  $InTsTmDv$ , блок параметрів  $PbTsTmDv$ , коментар  $CmTsTmDv$ .

Правила вибору визначають неформальні переваги, які треба врахувати під час вибору залежно від обраного типу прототипу та інших ознак, які загалом визначають сигнатуру ситуації спрацювання правила вибору. Всі ознаки беруть з параметрів специфікації. Сигнатура може містити всі параметри всіх специфікаційних об'єктів. Умова, яка визначається сигнатурою, змінює ступінь придатності варіантів – кандидатів.

Правила вибору утворюють певну множину правил:

$$LiRuChDv = M(RuChDv).$$

Кожне правило  $RuChDv$ :

$$RuChDv = \{M(CdOb), M(CdDv), FnRuCh, CmRuCh\},$$

де  $M(CdOb)$  – множина умов щодо параметрів прототипу та інших об'єктів системи;  $M(CdDv)$  – множина умов щодо параметрів об'єктів пристроїв-кандидатів;  $FnRuCh$  – функція зміни показника придатності;  $CmRuCh$  – коментар.

Правила сумісності  $RuCf$  для певного шаблону компонента та іншого, вже обраного компонента, ставить у відповідність зміну показника придатності. При цьому можуть накладатися додаткові параметричні обмеження, визначається і тип зв'язку між двома компонентами (контейнерний, присднання, інший).

$$LiRuCh = M(RuCf)$$

$$RuCh = \{idRuCh, idTmDv, idTmDv', LDvDv, FnRuCh, CmRuCh\},$$

де  $idTmDv$ ,  $idTmDv'$  – ідентифікатори компонент, сумісність яких порівнюється;  $LDvDv$  – тип зв'язку між компонентами;  $FnRuCh$  – функція зміни придатності варіанта та параметричного обмеження;  $CmRuCh$  – коментар.

Розглянемо послідовність задач, які необхідно вирішити для вибору об'єкта-компонента для певного прототипу  $Pt_i$  з врахуванням даних проектування.

1. Для даного прототипу  $Pt_i$  з каталогу шаблонів компонентії вибирається множина об'єктів. Вибір здійснюється на підставі збігу типів об'єкта компонентії та прототипу.

2. Всі обрані об'єкти заносять у тимчасову таблицю  $LiChDv$ :

$$LiChDv = M(ChDv)$$

$$ChDv = \{idTmDv, TmDv', pref\},$$

де кожен варіант вибору  $ChDv$  складається з ідентифікатора шаблону вибору  $idTmDv$ , опису певного параметричного та структурного звуження цього шаблону  $TmDv'$  та нормованої оцінки ступеня придатності об'єкта  $pref$ .

3. Для кожного компонентійного об'єкта з цієї таблиці порівнюють діапазони параметрів з параметрами прототипу. Для цього з прототипом має бути асоційовано декілька функцій.

Функція оцінки придатності об'єкта-кандидата:

$$AccDev(PbPt, PbTmDv) \rightarrow pref$$

ця функція на підставі параметричних вимог прототипу та параметрів шаблону об'єкта компонентії визначає ступінь його придатності для вирішення поставлених задач. Оцінки придатності нормуються для всього списку обраних пристроїв.

Функція визначення придатної конфігурації створює певне звуження шаблону, яке задовольняє вимоги прототипу. Звуження визначається за конфігурацією, інтерфейсами та параметрами шаблону.

$$AccConfDv(PbPt, TmDv) \rightarrow TmDv'$$

Крім того, для вирішення задачі міжрівневого параметричного відображення має бути визначена функція

$$DefPtDv(PbPt) \rightarrow PbTmDv$$

$$DefDvPt(PbTmDv) \rightarrow PbPt.$$

За результатами визначення параметричної відповідності деякі варіанти вибору можуть бути відкинуті.

4. Для кожного елемента множини правил вибору перевіряється умова придатності правила. Ця умова стосується як всіх визначених специфікаційних об'єктів, так і об'єктів-кандидатів загалом. Якщо умова істинна, то з використанням функції  $FnRuCh$  відбувається перерахунок ступеня придатності всіх варіантів.

$$FnRuCh(M(CdOb), M(CdDv)) \rightarrow M(idTmDv, pref).$$

Задача врахування загальних правил вибору може вирішуватися один раз для всіх об'єктів-кандидатів.

5. Вирішується задача оцінки сумісності для кожного об'єкта-кандидата. Перевіряється його сумісність з об'єктом-контейнером та всіма приєднаними до нього об'єктами (якщо вони вже обрані). У результаті перевірки змінюється ступінь придатності варіанта та можуть бути обмежені конфігураційні та параметричні значення, що описується функцією:

$$FnRuCf: (TmDv, M(Dv)) \rightarrow (pref', PbTmDv').$$

Як правило, спочатку перевіряють на сумісність з контейнером, а потім – з приєднаними об'єктами. Уточнюються типи інтерфейсів.

6. На цьому етапі вибирається об'єкт-кандидат з найбільшим показником придатності. Якщо об'єкт вибрати не вдалося, то повертаються до попередніх етапів вибору, обирають об'єкт з меншим показником придатності і т.д. Процедура вибору може відбуватися і в діалоговому режимі, коли система пропонує варіанти, оцінки їх придатності та пояснює недоліки та переваги кожного варіанта вибору, а сам вибір вже здійснює проектувальник.

Побудовану мережу об'єктів компонентії використовують під час експлуатації інформаційної системи.

### *Задачі стадії експлуатації ІС*

На стадії експлуатації ІС працюють з усіма побудованими на попередніх етапах специфікаціями. Використання специфікацій вищих рівнів дає змогу відобразити реальні параметри функціонування та параметри виконання бізнес-процесів, застосувати наявні методи аналізу та оптимізації ІС, визначати вузькі місця не тільки у функціонуванні фізичної системи, а передусім у виконанні бізнес-процесів.

Під час експлуатації системи вирішується великий та складний комплекс задач аналізу, керування, налаштування та оптимізації. Сьогодні експлуатаційні задачі керування найповніше розв'язані для мережевої складової частини ІС. Тут діють загальновизнані та стандартизовані архітектури керування, протоколи, розробляються нові концепції, які інтелектуалізують та автоматизують цей процес [2]. Водночас керування, аналіз та оптимізація на рівні систем і особливо застосувань значною мірою виконується вручну, інтуїтивно, без належного обґрунтування та документування цих процедур та комп'ютерної підтримки. Керування параметрами ІС на рівні бізнес-процесів знаходиться у зародковому стані. У літературі [3] підкреслюється необхідність вирішення задачі керування для інформаційної системи в комплексі. Вирішення цієї задачі вважається одним з першочергових для розвитку інформаційних технологій.

Сьогодні існує значна кількість систем керування інформаційними ресурсами корпоративної ІС [5]. Як правило, вони базуються на концепції інтелектуальних агентів керування та центральної платформи керування, яка може бути розподіленою. Головними об'єктами керування [6] є:

- локальна комп'ютерна мережа;
- телекомунікації (сполучення з глобальною мережею);
- системи;
- бази даних;
- бізнес-застосування;
- система інформаційної безпеки.

У [7] задача керування поділена на:

- керування мережами;
- керування системами;
- керування сервісами.

У [4] подано такий перелік задач керування інформаційними ресурсами:

- мережеве адміністрування (облік мережевих пристроїв, керування ними, багатопротокольність);
- моніторинг мережі та систем;
- керування навантаженням;
- опрацювання збійних ситуацій;
- керування системою архівування та міграції даних;
- обслуговування робочих груп (інсталяція ПЗ, віддалене керування, облік апаратного та ПЗ в різних групах, керування локальними серверами);
- адміністрування та віддалений моніторинг СУБД;
- інтегрована система безпеки даних (адміністрування користувачів, ідентифікація, авторизація, аудит);
- інтеграція та взаємодія з системами керування телекомунікаційним обладнанням.

Важливою компонентою такої системи керування є блок реагування на події. У [6], наприклад, розвинуто механізм опрацювання подій. Система дозволяє програмувати роботу групи агентів з моніторингу мережі залежно від подій, автоматизувати її, попередньо сформулювавши правила та завдання на моніторинг.

Ширшого застосування набуває і використання функцій штучного інтелекту та інженерії знань в системах керування інформаційними ресурсами. Наприклад, в [5] дані про застосування та процедури роботи з ними зберігаються в інтелектуальних агентів у вигляді спеціальних модулів знань (Knowledge modules). У [6] запропоновано використання Модуля керування проблемами для пошуку глибинних причин виникнення проблеми та її усунення. При цьому також використовуються знання.

Водночас сучасні системи керування інформаційними ресурсами є конгломератом продуктів та функцій, не пов'язаних між собою єдиною моделлю системи, єдиною політикою керування. Нерідко запропоновані інтелектуальні функції сформульовані лише концептуально, без вказання можливих механізмів їх реалізації. Найвищим рівнем керування в них є керування бізнес-застосуваннями, які є тільки складовою частиною компоненти бізнес –процесу. Загалом, наявні сьогодні системи керування інформаційними ресурсами досить розвинуті функціонально і становлять непоганий фундамент для розробки нових генерацій систем керування, для яких буде характерне єдине цілісне подання та розуміння ІС, широке використання інженерії знань для вирішення задач, вирішення задач не тільки керування та налаштування, але й оптимізації, зміни структури, перепроєктування, динамічної адаптації до змін бізнес-процесів.

Крім суто функціональної класифікації, задачі етапу експлуатації ІС можна класифікувати, наприклад, за ознакою частоти та вимог до часу вирішення. При цьому можна виділити:

- задачі забезпечення визначеного рівня обслуговування застосувань та оперативне (автоматичне) реагування шляхом зміни параметрів;
- задачі адаптації системи до зміни бізнес-процесів (БП змінюються рідше, виявлення таких змін вимагає аналізу моделі БП, що, в свою чергу, вимагає більших витрат часу);
- задачі аналізу та оптимізації системи, пошук вузьких місць з врахуванням структури БП;
- модернізація, придбання більш продуктивного обладнання;
- перепроєктування системи, зміна архітектурних вирішень різних рівнів.

Крім того, всі задачі рівня експлуатації, що розглядаються нами, можна поділити на:

- задачі, спрямовані на формування та підтримку системи проектування;
- задачі, спрямовані на роботу з самою ІС із використанням системи проектування (аналіз, оптимізація, керування, координація, проектування). Такі задачі будемо називати прикладними задачами проектування.

Задачі моніторингу та мережеметрії належать до класичних задач діагностики та керування в ІС. У сучасних системах завдання для моніторингу (які параметри вимірювати, як вимірювати, як часто вимірювати) задає адміністратор системи. Вони охоплюють нижчий рівень аналізу системи. Метою моніторингу найчастіше є контроль за параметрами працездатності окремих пристроїв та систем. Завдання на моніторинг ніяк не обґрунтовуються. Пізніше результати моніторингу узагальнюються статистично процедурами мережеметрії.

З використанням системи формальних специфікацій проектування можна зробити задачу моніторингу та мережеметрії гнучкішою, інтелектуальнішою, більш обґрунтованою. Головні відмінності такої процедури від традиційної:

- завдання інтелектуального моніторингу впливають з вищого рівня абстракції у поданні системи. Досліджуються не параметри функціонування конкретного пристрою, а параметри виконання застосування або бізнес-процесу. Таке завдання є складнішим, оскільки може передбачати вимірювання цілого комплексу параметрів. Водночас вимірювання параметрів виконання бізнес-процесу є більш ефективним, адже розглядає тільки ті параметри систем, які впливають на параметри виконання БП;
- з використанням параметричних залежностей можна обмежити набір параметрів моніторингу і так зменшити навантаження системи;
- можна визначати параметрично ситуації та події у системі, які визначають запуск конкретної задачі моніторингу, що зробить реакцію на подію оперативнішою.



Формально моніторинг описується як список завдань на моніторинг.

$$LiMn = M(Mn).$$

Завдання на моніторинг Mn, крім ідентифікатора завдання idMn, містить умову виконання завдання CnMn, процедуру моніторингу ScMn, ReMn – місце збереження результатів, коментар CmMn.

$$Mn = \{idMn, CnMn, ScMn, ReMn, CmMn\}.$$

Задачі інтелектуального моніторингу тісно пов'язані з задачами визначення параметрів об'єктів системи проектування. Формальне визначення параметра (Д.1) містить процедуру визначення параметра:

$$Pr = \{idPr, ZnPr, DnPr, idVm, DfPr, CmPr\},$$

де DfPr – процедура визначення параметра. Ця процедура може відображати:

- ручне визначення;
- визначення через параметричну залежність від інших параметрів;
- завдання на визначення параметра з файлів моніторингу та мережеметрії.

Процедури мережеметрії – це окремі процедури опрацювання результатів моніторингу, які зберігаються та описуються аналогічно процедурам моніторингу. Відмінність полягає в тому, що подією виконання процедури мережеметрії є завершення певної конфігурації завдань моніторингу. Результати мережеметрії також зберігають у файлах.

$$LiMm = M(Mm)$$

$$Mm = \{idMm, CnMm, ScMm, ReMm, CmMm\}.$$

Обов'язковою складовою частиною файла результатів моніторингу та мережеметрії є часова позначка. Це дозволяє проводити додатковий моніторинг, якщо параметри попереднього застаріли.

Завдання на моніторинг запускаються автоматично при настанні певної події у системі, умова якої формулюється параметрично над параметрами функціонування та специфікаційними параметрами об'єктів всіх рівнів або за потреби вирішення прикладних задач системи проектування та моделювання.

### Висновки

Формальні специфікації для проектування мережі компонент системи можуть бути покладені в основу розвинутих систем проектування та керування складними розподіленими системами, що в свою чергу призведе до підвищення якості їх функціонування, можливості швидкої адаптації до змін умов реалізації процесів підприємства.

1. Буров Є.В. Система формальних специфікацій для проектування розподілених інформаційних систем // Вісник Державного університету "Львівська політехніка". – 2000. – № 406. – С.50–59.
2. Иванов П. Активные сети // Сети. – №10. – 1999.
3. Штайнке С.В. Поиск интегрированного управления // LAN. – №1. – 1999.
4. Кутукова Е. Системы управления информационными ресурсами // Открытые системы. – №3. – 1998. – С. 24.
5. Семенов М. Система PATROL – управление распределенными ресурсами // Открытые системы. – №3. – 1998. – С 26.
6. Черный В. TransView – управление распределенной корпорацией // Открытые системы. – №3. – 1998. – С. 31.
7. Катыев С. Об одной концепции управления распределенными ресурсами // Открытые системы. – №3. – 1998. – С. 36