ОБСЛУГОВУВАННЯ АВІАЦІЙНОЇ ТЕХНІКИ

УДК 629.735.083:004.942(045)

О.А. Тамаргазін, Хаммуд Нізар

МЕТОДИКА ОЦІНКИ ТЕХНІЧНОГО СТАНУ АВІАЦІЙНОЇ ТЕХНІКИ З ВИКОРИСТАННЯМ МОДЕЛЮВАННЯ НА БАЗІ АГРЕГАТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Розглянуто використання агрегативної побудови програмного забезпечення інформаційної підтримки експлуатації авіаційної техніки при моделюванні образу технічного стану підконтрольного об'єкта.

При розробці програмного забезпечення для оцінки технічного стану авіаційної техніки вирішуються питання, пов'язані з організацією збереження й обслуговування інформації, яку можна умовно розділити на оперативну й довідкову.

Довідковою інформацією є описи модулів, оформлених як агрегати (включаючи методи й елементи зовнішнього програмного забезпечення). Із цих модулів на підставі оперативної інформації, отриманої у діалозі з користувачем, будується моделююча програма, що враховує особливості конкретного виробу, технічний стан якого оцінюється.

Уся довідкова інформація системи зберігається в реляційних базах даних, які використовуються для збереження вихідних текстів програм і макровизначень – бібліотеки агрегатів, які можуть містити один або кілька елементів. Під елементом розуміють опис агрегату, методу або будь-якого іншого компонента довідкової інформації. Довідкова система може містити не одну, а декілька бібліотек агрегатів.

Структурно бібліотека агрегатів є записом у базі даних, який містить опис одного або декількох елементів. Кожний елемент складається з двох частин: заголовка і тіла. Заголовок містить ім'я елемента і, якщо необхідно, коментар. Тіло елемента розташовується після заголовка і також може містити коментарі.

При створенні автоматизованої інформаційної системи (АІС) оцінки технічного стану авіаційної техніки були використані такі пропозиції, що обумовили структуру внутрішнього програмного забезпечення

1. Елементи, з яких компонується модель, і модулі зовнішнього програмного забезпечення представлені у вигляді агрегатів, що, зокрема, ліквідує проблеми, пов'язані з поповненням і модифікацією бібліотеки методів. Крім того, це дає можливість всередині АІС об'єднати два замовлення – на модель і на аналіз експлуатаційної інформації Програми, які підтримують процес імітації, також частково будують за агрегативним принципом. Таке представлення уніфікує спільну роботу окремих програм, що полегшує їхнє тестування і дозволяє порівняно просто вносити необхідні зміни.

2. Елементи моделі і зовнішнього програмного забезпечення, необхідні для проведення аналізу експлуатаційних даних, на основі об'єднаного замовлення збирають в єдину систему (метасистему), яка функціонує за законами агрегативних моделей.

3. Введено поняття схеми еквівалентності, що дає можливість оголошувати тотожними деякі змінні в метасистемі, тобто змінні власне моделі і використовуваних методів. Це рішення дозволяє простими засобами здійснювати знімання даних з моделі для їхньої обробки.

4. У процесі моделювання по черзі запускається досліджувана модель і зовнішнє програмне забезпечення. У термінах метасистеми це означає можливість (і необхідність) зупинки "системних годинників" для однієї її підсистеми і запуску їх для іншої підсистеми. Тому внутрішнє програмне забезпечення передбачає наявність окремих годинників для різних підсистем агрегативної моделі.

5. Введено поняття пріоритету сигналів, які циркулюють у метасистемі. Це дає можливість організувати спільну роботу зовнішнього програмного забезпечення з моделлю та необхідний порядок обробки сигналів усередині моделі.

6. Передбачено також ряд технічних рішень, наприклад, індексування "входів" і "виходів" агрегатів для зручної адресації сигналів.

Роботу моделюючої програми можна подати як повторюваний ряд кроків:

1) з упорядкованого за модельним часом і пріоритетом списку подій вибирається чергова подія;

2) модельному часу присвоюється значення часу з інформаційного поля запису події;

3) призначається процедура, яку необхідно виконати в даний момент модельного часу;

4) знайдена в попередньому пункті процедура "запускається" на виконання.

Результатом виконання зазначеної процедури є перетворення дерева метасистеми і деякої кількості вихідних сигналів, які заносяться в список подій. Далі кроки 1 – 4 повторюються знову. Робота моделюючої програми закінчується або після вичерпання списку подій планувальника, або за командою про закінчення роботи.

Таким чином, оцінку технічного стану конкретного виробу авіаційної техніки в АІС можна подати як сукупність задач аналізу моделей, логічно об'єднаних у рамках замовлення на дослідження.

Вузловим поняттям аналізу даних є задача, яка являє собою програмно реалізовану процедуру перетворення деяких вхідних даних **X**, що знімаються з моделей, у вихідні дані **Y**, які є результатом виконання процедури. Якщо, наприклад, розглянути задачі звичайного статистичного оцінювання, то вхідними даними **X** є одержувана з моделі вибірка або вибіркова траєкторія, а вихідними даними **Y** – оцінюваний показник (середньоквадратичне, дисперсія, гістограма тощо) [1]. Проблем з передачею даних з моделі для їхньої обробки не виникає, якщо сукупність **X** збігається із сукупністю траєкторій деякого набору координат (станів) моделі, оскільки у момент появи подій в агрегативній моделі, коли змінюються координати із зазначеного набору, відповідно змінюються і дані **X** .

У реалізованому варіанті АІС як вхідні дані можна розглядати послідовності та функції, обумовлені такими поняттями і конструкціями [2; 3].

Нехай **Z** – простір станів досліджуваної агрегативної моделі, $z(t)$ – її стан у момент часу t, $\mathbf{Y} \subset \mathbf{Z}$ – фіксована підмножина простору станів, $\mathbf{E} = \{e_i\}$ – послідовності подій, які відбуваються в агрегативній системі і збігаються з моментами $\mathbf{T} = \{t_1, t_2, \ldots\}$ стрибкоподібних змін її координат. У послідовності **E** можна виділити підпослідовності $\mathbf{E}^{(i)}$, які не перетинаються, побудовані для моментів $\mathbf{T}^{(i)} = (t_1^{(i)}, t_2^{(i)}, \ldots)$ стрибкоподібних змін i-ї координати і $\mathbf{T}^{(i_1,\ldots,i_n)} = \mathbf{T}^{(i_1)} \cup \ldots \cup \mathbf{T}^{(i_n)}$.

Введемо такі підпослідовності особливих моментів часу:

$$\mathbf{T}_B = \mathbf{T} \cap \{t : z(t) \in \mathbf{B}\};$$

$$\mathbf{T}_B^{(i)} = \mathbf{T}^{(i)} \cap \mathbf{T}_B;$$

$$\mathbf{T}_B^{(i_1,\ldots i_n)} = \mathbf{T}^{(i_1,\ldots i_n)} \cap \mathbf{T}_B,$$

побудовані для моментів перебування траєкторії моделі в підмножині **B**. Зафіксуємо набір $(i_1,\ldots,i_n)$, множину $\mathbf{Y} \subset \mathbf{Z}$ і сукупність дійсних функцій $f(z) = (f_1(z), \ldots, f_m(z)) : \mathbf{Z} \to \mathbf{R}^m$. Вхідними даними **X** є послідовність вигляду $\{(t, f(z(t))), t \in \mathbf{T}_B^{(i_1,\ldots,i_n)}\}$. Зокрема, якщо функція $f(z)$ тотожно стала, то ця послідовність збігається з послідовністю $\mathbf{T}_B^{(i_1,\ldots,i_n)}$.

Таким чином, для того, щоб задати вхідні дані за допомогою описаної конструкції, потрібно задати набір $(i_1,\ldots,i_n)$, множину $\mathrm{Y} \subset \mathrm{Z}$ і набір дійсних функцій $f(z)$.

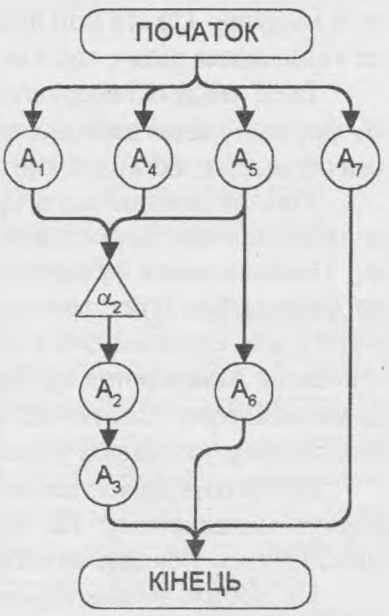
Алгоритм, який реалізує ту чи іншу задачу аналізу, перетворить дані X у вихідні дані Y, за які в АІС прийняті дійсні числа або вектори. При цьому сам алгоритм може характеризуватися ще набором дійсних параметрів P.

Кожна задача забезпечується ознакою δ її закінчення. У процесі аналізу даних задача може завершуватись результативно в тому значенні, якщо вихідні дані Y отримано, або не результативно, якщо вихідні дані не отримано. Останнє може відбутися, наприклад, якщо досліджувана подія не настає або отримані вихідні дані не є представницькими. Будемо вважати, що зазначена ознака приймає значення $\delta = 1$ у момент завершення роботи алгоритму, що реалізує відповідну задачу, а до цього моменту він дорівнює 0.

Об'єднання задач в експеримент здійснюється таким чином.

Нехай у процесі аналізу даних необхідно вирішувати задачі A_1, …, A_N, … . Кожна з цих задач характеризується відповідним набором $(X_1, Y_1, P_1, \delta_1),\ldots,(X_N, Y_N, P_N, \delta_N)$, … . Сукупність перерахованих задач можна зобразити у вигляді сіткового графіка (див. рисунок). При цьому деякі задачі є первинними – початок їхнього рішення не обумовлюється іншими задачами.

Необхідність рішення (а отже, і момент початку рішення) інших (вторинних) задач обумовлюється фактом закінчення рішення інших задач. Наприклад, задача A_2 може бути вирішена лише за умови вирішення задач A_1, A_4, A_5, A_7, задача A_3 вирішується після рішення задачі A_2, а A_6 – після рішення задачі A_5. Така організація аналізу даних дає можливість "ощадливого" включення тієї чи іншої задачі, тобто у випадку, якщо в її рішенні дійсно є необхідність. Якщо задача A_i є вторинною і на необхідність її рішення впливають задачі $A_{i_1},\ldots,A_{i_n}$, то в АІС вважається, що рішення задачі A_i може початися в момент, коли її логічна функція $\alpha_i = \alpha_i(\delta_{i_1},\ldots,\delta_{i_n})$ прийме значення 1. Зокрема, якщо $\alpha_i = \delta_{i_1} \wedge \ldots \wedge \delta_{i_n}$, то задача A_i може вирішуватись лише після того, як будуть вирішені всі попередні їй задачі $A_{i_1},\ldots,A_{i_n}$. Якщо ж $\alpha_i = \delta_{i_1} \vee \ldots \vee \delta_{i_n}$, то задача A_i може вирішуватися лише після того, як буде вирішена хоча б одна з задач $A_{i_1},\ldots,A_{i_n}$.



Приклад експерименту

Таким чином, для аналізу експлуатаційних даних з оцінки технічного стану авіаційної техніки необхідно:

– задати сукупність розв'язуваних задач $A_{i_1},\ldots,A_{i_n}$ разом з наборами вхідних даних $X_1,\ldots,X_n$ і параметрами $P_1,\ldots,P_n$;

– виділити в даній сукупності первинні й вторинні задачі;

– для кожної вторинної задачі A_i задати логічну функцію α_i, яка визначає можливість початку її рішення.

Перелік задач, розв'язуваних у процесі аналізу експлуатаційних даних, невеликий, і самі задачі досить стандартні – оцінки стаціонарних характеристик, перевірки гіпотез, регресійний аналіз тощо. Таким чином, задача має стандартну обчислювальну частину (власне алгоритм перетворення даних), який називаємо методом рішення задачі. Цей метод застосовується до даних **X** для одержання даних **Y**, причому вхідні дані **X** стандартними вже ніяк не будуть. Більш того їхня різноманітність визначає різноманітність задач дослідження, але та сама задача може бути вирішена різними методами.

Описаний вище граф задає лише логічну послідовність рішення задач. Реальна ж послідовність їх рішення (у часі) узгоджується з логічною послідовністю і визначається ще можливістю або неможливістю одночасного рішення задач. Інформація про "сумісність" задач має бути відомою і вказуватися, наприклад, у паспортних даних методів. Ця інформація користувачу недоступна і використовується програмами зовнішнього програмного забезпечення для задання належної послідовності виконання задач.

Зовнішнє програмне забезпечення служить для реалізації експериментів подібних описаному типу з агрегативними моделями. Вихідною інформацією для цього є замовлення на експеримент (аналіз експлуатаційних даних), який формується користувачем за допомогою діалогової системи й утримує відомості про особливості аналізу даних за конкретним типом авіаційної техніки (набір розв'язуваних задач, методів рішення, вхідних даних, інформація про взаємозв'язок задач). Власне зовнішнє програмне забезпечення складається з двох частин: бібліотечних модулів, які реалізують різні методи дослідження моделей; виконавчих модулів, які реалізують проведення експерименту відповідно до вимог користувача.

Модулі-методи є консервативною частиною зовнішнього програмного забезпечення. Їхній склад визначається лише поточним станом методів дослідження складних систем, про-

блемною орієнтацією і не залежить від типу досліджуваної авіаційної техніки. Навпаки, виконавчі модулі повинні бути погоджені як з проведеним дослідженням, так і з типом досліджуваної техніки, тобто здатними перебудовуватись.

Усього є чотири виконавчих модулі M_1, …, M_4. Модуль M_1 називається керуючою програмою. Він характеризується постійною структурою, тобто не змінюється при зміні завдання й модулів. Проте при функціонуванні модуля M_1 у ньому використовується граф черговості включення задач, що є по суті параметром модуля M_1.

Інші модулі генеруються на основі замовлення користувача і виконують такі функції: M_2 формує вхідні дані для задач; M_3 запускає відповідну задачу; M_4 обчислює значення логічних функцій, які визначають можливість початку запуску задач.

Робота зовнішнього програмного забезпечення здійснюється в кілька етапів. Спочатку за замовленням користувача генеруються модулі M_2, M_3, M_4, задаються параметри модуля M_1, викликаються бібліотечні модулі-методи, необхідні в даному експерименті. З цих модулів формується агрегативна система зі стандартною структурою. Далі встановлюється еквівалентність між станами даної системи і станами досліджуваної моделі. Вся отримана інформація поповнює замовлення на модель, яка перетворюється в такому разі в об'єднане замовлення на метасистему. Зазначена генерація й поповнення замовлення виконуються допоміжними програмами, які надалі участі в моделюванні не приймають.

Об'єднане замовлення передається у внутрішнє програмне забезпечення, яке власне і формує метасистему. На наступному етапі робота зовнішнього програмного забезпечення здійснюється у складі метасистеми.

Ця робота виконується так. При настанні події у моделі на вхідну клему агрегату M_2 надходить сигнал, який містить повні імена координат моделі, що змінилися в цей момент стрибкоподібно. Агрегат M_2 аналізує, чи впливають ці зміни на рішення якої-небудь задачі, і у випадку впливу на відповідний агрегат-метод подається необхідний вхідний сигнал для подальшої обробки. Вказаний вхідний сигнал формується на основі набору $(i_1,\ldots,i_n)$, множини $\mathbf{B} \subset \mathbf{Z}$ і функції $f(z)$, що задають цей сигнал і визначили вид агрегату M_2 при його генерації.

Таким чином, здійснюється збір і обробка даних у процесі моделювання.

Виконання задач відповідно до встановленого розкладу відбувається таким чином. На початку моделювання агрегат M_1 за виглядом графа виконання задач визначає, які задачі можна почати вирішувати відразу ж. Далі здійснюється перевірка можливості паралельної роботи обраних задач. Визначаються задачі, які будуть вирішуватись, і на агрегат M_3 видається сигнал, який містить імена відповідних методів. У свою чергу агрегат M_3 запускає методи в роботу.

Після завершення роботи методу (якщо ознака, що характеризує задачу, прийняла значення 1) на агрегат M_1 подається сигнал і за виглядом графа виконання задач визначається, які задачі пов'язані з вирішеною. Сигнал з номером кожної такої задачі надходить до агрегату M_4, в якому обчислюється значення логічної функції, що дозволяє запуск задачі. Якщо запуск можливий, то інформація про це повідомляється агрегату M_1, і далі сам запуск здійснюється описаним вище способом за допомогою агрегату M_3.

У процесі роботи програмного комплексу фіксуються вихідні дані, отримані при рішенні задач, а також факт нерозв'язності тієї чи іншої задачі.

Список літератури

1. *Дубровский Н.Г., Мокроус М.Ф.* Параметрические методы диагностического контроля состояния авиадвигателей. Линейные диагностические матрицы // Тр. ЦИАМ. – М.: 1981. – № 964. – 28 с.

2. *Douglas R. Shier, K. T. Wallenius*. Applied Mathematical Modeling. A Multidisciplinary Approach. CRC Pr. –1999. – Nov. – P. 472.

3. *Edward B. Saff, Arthur D. Snider.* Fundamentals of Complex Analysis for Mathematics // Science and Engineering – 2 ed. (Febr. 10, 1993) Prentice Hall. – P. 528.

Стаття надійшла до редакції 2 жовтня 2000 року.