відміну від get методу дає змогу приховати усі дані, які передаються в тіло запиту, а не в заголовок, що потім відображається в браузері, і легко може бути зчитаний зловмисником.

Висновок

На основі принципів системного аналізу створено концептуальну модель предметної сфери «Захист аудіоінформації». Створена АСОІ з обмеженим доступом засобами реляційної бази даних, СКБД MSSQL та мови програмування Java. Система є програмним продуктом у вигляді веб-сайту, який реалізований на основі клієнт-серверної архітектури.

1. Електронний ресурс. Режим доступу: www.defend-seminars.ru/clause/clause/16466/1945/. 2. Классификация микрофонов в зависимости от предназначения. Режим доступу: www.pulscen.ru/info/special/specialequipment/microphone/application. 3. Способы и средства подавления устройств несанкционированного перехвата информации с телефонных линий / А. Хорев. – Режим доступу: www.bre.ru/security/19053.html. 4. Способы и средства защиты информации / А. Хорев. – Режим доступу: www.analitika.info/zaschita.php. 5. Соболев А.Н. Физические основы технических средств обеспечения информационной безопасности / А.Н. Соболев, В.М. Кириллов. – М.: Гелиос АРВ, 2004. – 224 с. 6. Хорев А.А. Защита информации от утечки по техническим каналам. – Ч.1: Технические каналы утечки информации: учеб. пособ. – М.: Гостехкомиссия России, 1998. — 320 с.

УДК 681.322

А.Б. Керницький
Національний університет “Львівська політехніка”,
кафедра систем автоматизованого проектування

ІНТЕГРАЦІЯ ЗНАНЬ У САПР ТП ІЗ ВИКОРИСТАННЯМ ВИСОКОРІВНЕВИХ МЕРЕЖ ПЕТРІ

Наведено інкрементаційний метод для побудови динамічного опису виробу та процесів його виготовлення у САПР ТП із використанням високорівневих мереж Петрі для інтеграції знань, поданих у вигляді продукційних правил і фреймів. Запропоновано методику вирішення конфліктів у моделях, побудованих із використанням високорівневих мереж Петрі.

Ключові слова – інтеграція знань, високорівневі мережі Петрі, САПР ТП.

The work presented incremental method for dynamic description of the product and its manufacturing processes in CAD TP using high-level Petri nets for integration of knowledge represented as production rules and frames. The method of resolving conflicts in models has been built using high-level Petri nets.

Keywords – knowledge integration, high-level Petri nets, CAD, CAM.

Вступ

Останнім часом було запропоновано кілька підходів до використання технологій штучного інтелекту для опису моделей виробу та керування виробничими процесами його виготовлення [1–4]. Особливої популярності для подання декларативного опису поведінкових знань предметної області у базах знань набули продукційні правила. Проте виявилося, що продукційні правила не є адекватними для визначення термінів, опису виробничих об’єктів і взаємозв’язків між ними [5, 6]. Рішення були запропоновані у комерційних середовищах штучного інтелекту (LOOPS (Lisp Object-Oriented Language) або KEE) у вигляді мови інтеграції фреймів та продукційних правил для формування можливостей гібридного подання [7, 8].

Проте специфікація і моделювання логіки керування у дискретних системах, таких як виробничі системи, є дуже складною проблемою [9]. Наш концептуальний підхід до моделювання конструкції виробу полягає у інтеграції формальних технологій, таких як високорівневі мережі Петрі (ВРМП) із моделями, що ґрунтуються на знаннях. Головною метою цієї стратегії є використання добре відомих можливостей мереж Петрі для опису дискретної поведінки (стани, переходи, перед- і постумови, синхронізація і паралельність) та їхні графічні можливості [10–13], та інтегрувати їх з іншими представленнями та з іншими технологіями логічного виведення, такими як системи, що ґрунтуються на знаннях і фреймах. Ми пропонуємо підхід до використання ВРМП для визначення поведінки технологічних процесів як дискретної системи і вони реалізовані у вигляді правил. Ми створюємо динамічні об'єкти, механізм наслідування яких використовується для синхронізації об'єктів. Стратегія використовується для знаходження варіантів розв'язання та інкрементаційного підходу, який дає змогу їх вирішити.

Основна частина

Загалом кожна модель системи виробництва є індивідуальною або класом виробничих сутностей (вироби, верстати, комори, транспортні пристрої тощо), що подаються у вигляді фреймів, які ми переважно називаємо об'єктами. Об'єкт може містити значення, зв'язки, знання і активні значення (демони). У доповненні до цих програмних властивостей, які підтримуються іншими фреймо- та об'єктно-орієнтованими мовами, наш підхід до моделювання включає набір семантичних примітивів, які реалізують формалізм, ґрунтується на ВРМП. ВРМП використовуються як формальний механізм для подання поведінкової інформації динамічної сутності з точки зору дискретної ситеми. Вузлами ВРМП подаються спеціалізовані слоти, які називаються слотами стану. Переходи ВРМП подаються правилами, а дуги – засновками та висновками правил. Кольори у вузлах визначаються об'єктами, які зберігаються у слотах станів. Двигун виведення системи правил визначає динамічну поведінку.

Слідуючи методології об'єктно-орієнтованого програмування, процес проектування починається із створення класу ієрархій, чиї елементи будуть у подальшому проілюстровані для побудови моделі виробничої системи. Три основні ієрархії використовуються для роботи із трьома різними аспектами: ресурси (фізичні), операції (функціональні) і вироби.

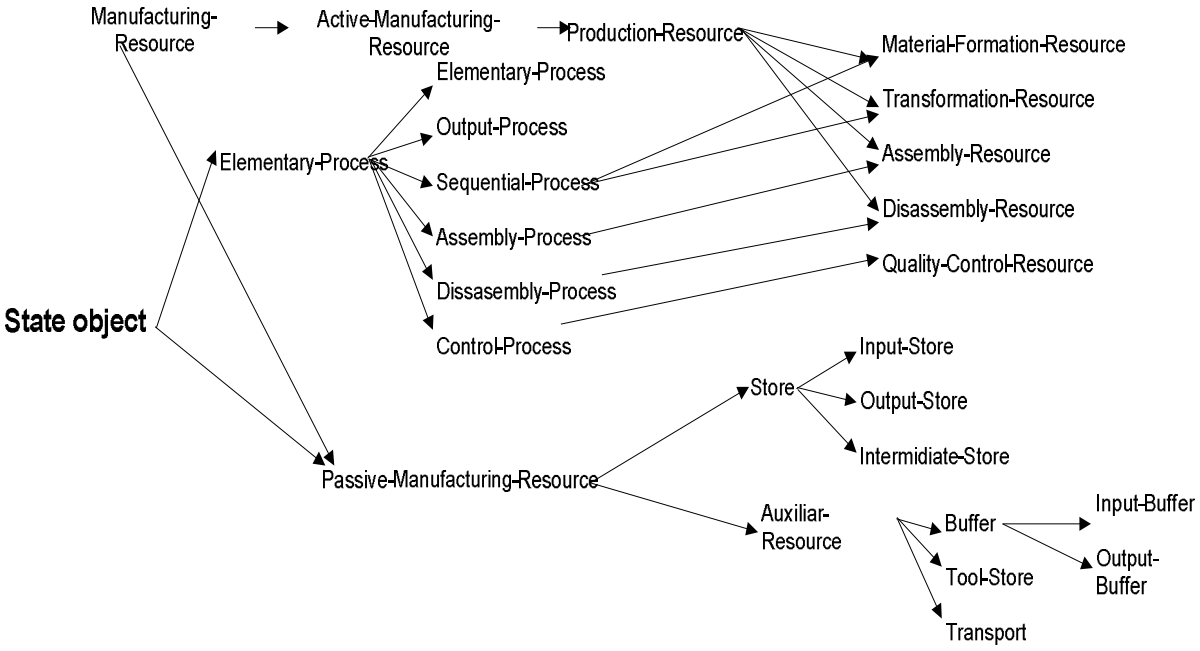

Рис. 1. Частина ієрархії виробничих ресурсів

Щоб зрозуміти доступний механізм моделювання для створення ізольованих сутностей, сфокусуємо свою увагу на поданні об'єктів ієрархії виробничих ресурсів (Transformation-Resources

(Трансформації-Ресурси)), показаних на рис. 1. Цей об'єкт містить прототипні знання про загальні ресурси перетворення. Він містить зв'язки, які визначають сутність у абстрактній ієрархії у моделі системи ((рівень-точності) precision-level (володіє-ресурсами) has-resources і (ресурс-чого) resource-of).

Він містить слоти для зберігання передбачуваної інформації (наприклад, доступні-об'єми available-capacity і pending-operations (відкладених операцій) для використання планувальником), для колекцій даних (наприклад, зберігати інформацію для функціональної статистики), для опису фізичних параметрів (наприклад, час виконання операцій), а також зберігає знання про поведінку як дискретну сутність події. Рис. 2 ілюструє графічне подання цієї поведінки, використовуючи графічні можливості ВРМП. Вузли завантаження, виконання, вивантаження у процесі і об'єм подані їхніми відповідними слотами станів у об'єкті Transformation-Resources (Трансформації-Ресурси).

Переходи розпочати-операцію, завершити-операцію і завершення-процесу реалізовані у вигляді правил, які визначені у відповідних слотах дій. Рис. 3 являє собою зовнішнє подання деяких з цих правил, які були реалізовані в КЕЕ. Що типово для об'єктно-орієнтованих систем, то це частина поведінки Transformation-Resources (Трансформації-Ресурси), яка успадковується від його предка, у цьому разі – від Sequential-Process (Послідовний-Процес).

Як тільки стає доступним клас ієрархії, користувач може вибирати сутності, які необхідні для побудови моделі системи. Наступним кроком є встановлення зв'язків між об'ктами, між фізичними ресурсами для побудови топології ресурсів підприємства, між операціями для побудови різних технологічних маршрутів. І на основі топології ресурсів підприємства і технологічних маршрутів отримують динамічну поведінку усієї системи.

Динамічні сутності зв'язку встановлюються засобами синхронізування переходів. Для побудованої моделі були необхідні два типи синхронізації: звичайна і двостороння. У результаті нормальної синхронізації відбувається заміна переходу іншим переходом із передумовою і висновком переходів, які синхронізуються (початковий перехід зникає), подальша синхронізація переходів здійснюється на основі цієї. Двостороння синхронізація нового переходу здійснюється із передумовами і висновками переходів, які синхронізуються (один початковий зберігається) і подальша синхронізація не залежить від нього.

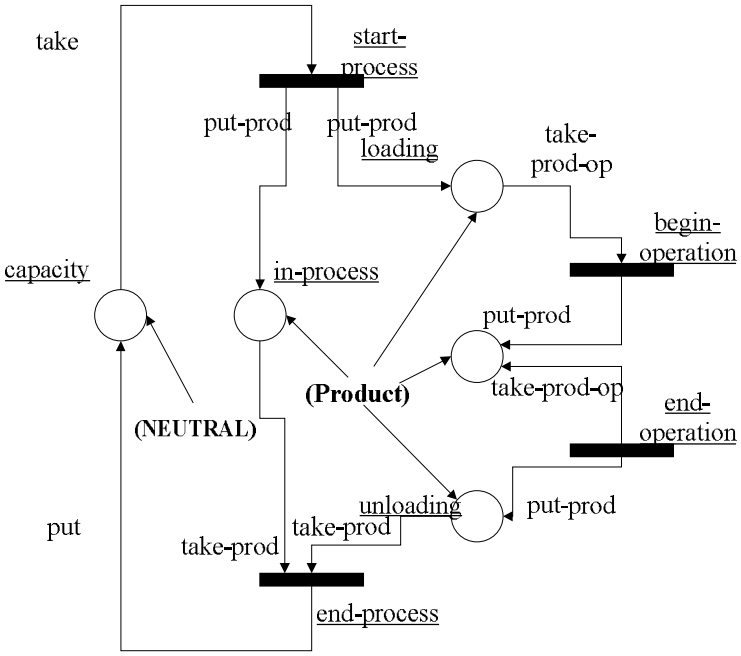

Рис. 2. Прототип Transformation-Resources

Правила можна подати як об'єкти у середовищі абстрактної ієрархії. Наслідуючи об'єктно-орієнтований підхід, синхронізацію можна реалізувати, використовуючи властивості наслідування правил. У разі нормальної синхронізації відбувається заміна правила, яке являє собою перехід до

іншого правила, предками якого є синхронізовані переходи. Усі ці предки використовуватимуться для подальшої синхронізації. У разі двосторонньої синхронізації створюється нове правило з усіма пов'язаними предками, а у подальшому будуть використовуватися предки, які належать до початкового правила.

```
{START-PROCESS
   (IF (?PRODUCT == IN CLASS PRODUCTS)
   (CAPACITY OF TRANSFORMATION-RESOURCE = TRUE)
   THEN
   (CHANGE.TO (CAPACITY OF TRANSFORMATION-RESOURCE = FALSE))
   (IN-PROCESS OF TRANSFORMATION-RESOURCE === ?PRODUCT)
   (LOADING OF TRANSFORMATION-MACHINE == ?PRODUCT))
}
{BEGIN-OPERATION
   (IF (LOADING OF TRANSFORMATION-RESOURCE == ?PRODUCT)
   THEN
   (OPERATE OF TRANSFORMATION-RESOURCE == ?PRODUCT)
   (DELETE (LOADING OF TRANSFORMATION-RESOURCE == ?PRODUCT)))
}
```

Рис. 3. Опис правил, які являють собою переходи розпочати-операцію (BEGIN-OPERATION) і розпочати-процес (START-PROCESS)

Слідуючи аналогічному підходу, операції описуються об'єктами із вбудованими мережами Петрі. Поведінка типової операції перетворення подана простою послідовною мережею із трьома переходами – розпочати-операцію, завершити-операцію і завершити-процес і двох вузлів – пошук-ресурсів, у-процесі. Іншим варіантом прикладання розроблених правил на вищому рівні абстакції у виробничій системі, які ґрунтуються на ВРМП, є побудова технологічних маршрутів:

If part "Circle" and diameter ≤ 20 мм, then 43/08; 41/02; 41/20; 41/18; 47/12; 41/05; 41/09; 41/07; 41/11; 41/61; 41/62; 41;07; 41/09; 41/13; 41/71; 42

If part "Circle" and diameter from 20 till 45 мм, then 43/03; 41/02; 41/20; 41/18; 47/12; 41/05; 41/09; 41/07; 41/11; 41/61; 41/62; 41;07; 41/09; 41/13; 41/71; 42

If part "Circle" and diameter ≥ 45 мм, then 43/04; ; 41/02; 41/20; 41/18; 47/12; 41/05; 41/09; 41/07; 41/11; 41/61; 41/62; 41;07; 41/09; 41/13; 41/71; 42

If part "Sheet" and thickness δ≤ 0,5 мм, then 43/09; 41/20; 47/02; 47/08; 41/19; 41/61; 41/63; 41/08; 41/71; 42

If part "Sheet" and thickness from δ=0,5 till δ=16мм, then 43/05; 41/20; 47/02; 47/08; 41/19; 41/61; 41/63; 41/08; 41/71; 42

If part "Sheet" and thickness ≥ δ=16мм, then 43/23; 41/20; 47/02; 47/08; 41/19; 41/61; 41/63; 41/08; 41/71; 42

If in specification "ГОСТ", then 19;42

If in specification "ОСТ", then 19;42

If in specification "ТУ", then 19;42

If in specification "ДСТ", then 19;42

If in specification is "СК", then 42

If in specification after the designation is letter "К", then first and last operation: (20;42)

Рис. 4. Опис правил, які являють собою переходи між технологічними маршрутами

Розроблені правила зберігаються у базі даних. На рис. 4 показано таблицю БД, у якій містяться правила: tbl_routerules and tbl_hard_ruletypes.

RULE_ID	RULETYPE_ID	RULE_SIGN	RULE_NUMBER1	RULE_NUMBER2	ID1	ID2	ROUTE_LIST	RULE_MATERIAL
2	1	1	20	45	1	1	24,22,72,9,13,11,15,31,32,11,13,17,40,47	
3	1	2	45	0	1	1	73,6,24,22,72,9,13,11,15,31,32,11,13,17,40,47	
4	1	3	0.5	0	3	2	77,24,70,78,23,31,33,12,40,47	
5	1	1	0.5	16	3	2	79,24,70,78,23,31,33,12,40,47	
6	1	2	16	0	3	2	80,24,70,78,23,31,33,12,40,47	
7	2	0	0	0	1	0	1,47,2	
8	2	0	0	0	2	0	1,47	
9	2	0	0	0	3	0	1,47	
10	2	0	0	0	4	0	1,47	
11	3	0	0	0	0	0	47	
12	4	0	0	0	0	0	2,47	
13	5	0	0	0	13	0	47	
14	6	0	0	0	2	0	74	
1	1	3	20	0	6	1	75,6,24,22,72,9,13,11,15,31,32,11,13,17,40,47	
15	7	0	0	0	0	0	1,47	гума
16	7	0	0	0	0	0	1,47	картон
17	7	0	0	0	0	0	1,47	папір
18	5	0	0	0	6	0	1,47	
19	7	0	0	0	0	0	1,47	войлок
20	1	3	2	0	2	1	1,76,28,32,31,36,12,35,40,31,47	
21	1	5	2	0	2	1	1,6,32,31,28,32,31,36,12,35,40,31,47	
55	6	1	0	0	3	1	,1,6,10	

Рис. 4. Подання продукційних правил у таблиці tbl_routerules

На рис. 5 показано конструктор створення нових правил для формування технологічних маршрутів обробки деталей.

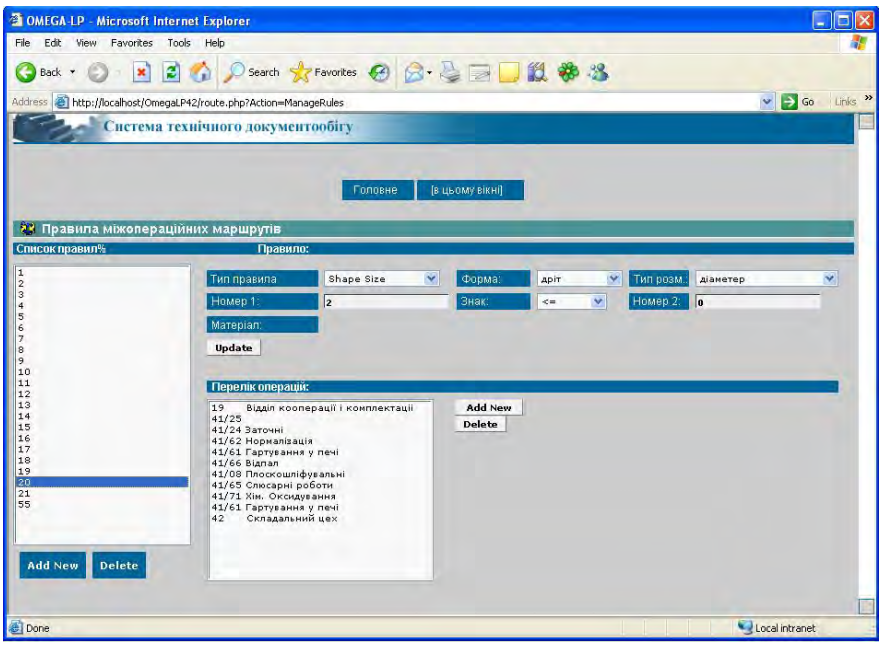

Рис. 5. Конструктор створення нових правил

Як було сказано раніше, динамічна поведінка об’єкта визначається відповідною мережею Петрі. Виникає одна проблема, пов’язана із недермінованою природою ВРМП. Ця недетермінована поведінка породжує проблеми прийняття рішень, які повинні бути вирішені. У середовищі виробничої системи більшість із них пов’язані із плануванням процесу прийняття рішень. Наприклад, якщо верстат може зберігати кілька доступних деталей, то може скластися необхідність у виборі однієї для її подальшої обробки.

У нашому підході проблеми прийняття рішень визначаються терміном “конфлікт”. Конфлікти – це спеціалізовані керуючі об’єкти, які містять інформацію про переходи із структурно

пов'язаними проблемами прийняття рішень з точки зору мережі, а також, як вирішувати проблему (політика керування пов'язана із конфліктом). Для того, щоб прийняти рішення на рівні експерта, необхідно класифікувати конфлікт властивостями планування проблеми, задіяними сутностями і процедурними цілями.

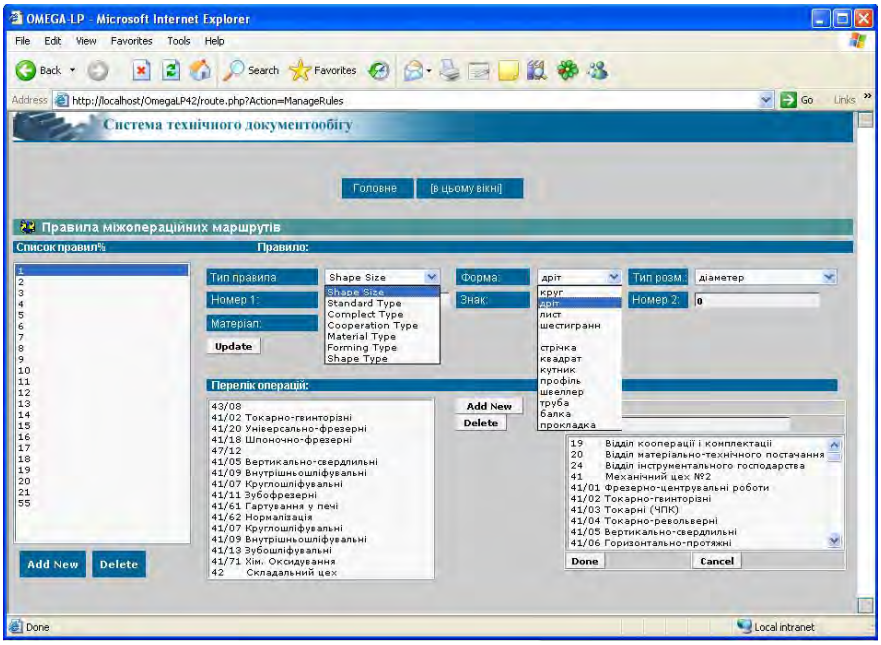

Рис. 6. Приклад формування правил

Висновки

Наведений інкрементаційний підхід для побудови динамічного опису системи можна використовувати для проектування політики прийняття рішень під час проектування техпроцесів у САПР ТП. Такий підхід хоча і не гарантує найкращого рішення, проте він є хорошим для автоматичного пошуку прийнятних рішень на початкових етапах проектування (під час прототипування), залишаючи оптимізацію на подальші етапи проектування.

Нові вузли рішення при побудові моделі з'являються із зв'язком між динамічними об'єктами. Кожна синхронізація – нормальна або двостороння – породжує новий конфлікт. Нормальна синхронізація генерує обмеженіший конфлікт, у той час, як двостороння генерує два взаємопов'язані конфлікти – один обмеженіший, а другий – пов'язаний із створенням нового шляху у ВРМП.

Політика контролю старих конфліктів підтримується і є пов'язаною із новими конфліктами. Отже, політику контролю можна звести до вибору між існуючими політиками контролю.

1. A.M. Flitman and R.D. Hurrion. Linking discrete-event simulation models with expert systems. Journal of the Operntions Research Society, 38:723-733, 1987. 2. Mark S. Fox and Stephen F. Smith. Isis: A knowledge-based system for factory scheduling. Expert Systems, 1(1):25–49, July 1984. 3. K. Jensen, Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Vol. 2, New York: Springer, 1995. 4. P. R. Muro-Medrano, J. Ezpeleta, and J.L Villarroel. A rule-petri net integrated approach for the modeling and analysis of manufacturing systems. In To appear in the Proceedings of the 13th IMACS World Congress. 5. Bobrow, D.G., and Stefik, M. J. Perspectives on Artificial Intelligence Programming. Science 231:4741, pp. 951–956, 28 February 1986. (Reprinted in Rich, C. & Waters R.C. (Eds.) Readings in Artificial Intelligence and Software Engineering, pp. 581-587, Los Altos: Morgan Kaufman Publishers, 1986. 6. G. Zhang and B.P. Zeigler. Artificial Intelligence, Simulation and Modeling, chapter The Systern Entity Structure: Knowledge Representation for Sirnulation Modeling and Design, pages 47–74. Wiley Interscience, New York, 1989. 7. Stefik, M.J., D.G. Bobrow, and K.M. Kahn. "Integrating Access-Oriented Programming into a Multiparadigm Environment." In: IEEE Software Vol 3, No. 1, January 1986. 8. Ward

Cunningham , Kent Beck, A diagram for object-oriented programs, Conference proceedings on Object-oriented programming systems, languages and applications, p.361-367, September 29-October 02, 1986, Portland, Oregon, United States. 9. Y.V. Reddy, M.S. Fox, N. Husain, and M. McRoberts. The knowledge-based simulation system. IEEE Software, pages 26-37, March 1986. 10. Murata T. Petri nets: Properties, analysis and applications. Proceedings 01 the IEEE, 16(1):39–50, January 1990. 11. J.L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, Englewood Cliffs, N.J., 1981. 12. C. Ramachandani. "Analysis of asynchronous concurrent systems by timed Petri nets," Technical Report MAC TR 120, MIT, 1974, Cambridge. 13. F.D.J. Bowden, "A brief survey and synthesis of the roles of time in Petri nets," Math. Comput. Modeling, Vol. 31, 2000, pp. 55–68.

УДК 004.9

М.Р. Мельник, П.Ю. Денисюк
Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування

АВТОМАТИЗАЦІЯ ПОБУДОВИ ГРАФІЧНИХ ЗАЛЕЖНОСТЕЙ У СИСТЕМІ ANSYS

Розроблено алгоритм автоматизованої побудови графіків у системі ANSYS з використанням постпроцесорів POST1 та POST26.

Ключові слова – автоматизація, ANSYS, постпроцесор, POST1, POST26.

Presented algorithm automation of graphical dependencies in ANSYS environment using the postprocessor POST1 and POST26.

Keywords – automation, ANSYS, postprocessors, POST1, POST26.

Вступ

Однією з найвідоміших систем для автоматизованого проектування МЕМС-пристроїв на компонентному рівні є ANSYS [1]. Процес аналізу задач проектування в ANSYS включає такі етапи [2–5]: 1) побудову моделі конструкції об'єкта проектування (ОП); 2) розв'язання задачі реакції ОП на різні фізичні дії; 3) постпроцесорну обробку вихідних результатів моделювання. Доступ до результатів отримують (залежно від типу отриманих даних) з використанням двох постпроцесорів POST1 і POST26. Система ANSYS дає змогу використовувати як графічний інтерфейс користувача, так і командний режим. Розроблення підсистеми автоматизованого збереження результатів і побудови графіків істотно спростить опрацювання та аналіз результатів структурного аналізу термоактюаторів, акселерометрів та інших МЕМС-пристороїв, в яких присутні механічні елементи. Під час проведення структурного аналізу важливу інформацію мають максимальні напруження і переміщення конструкції аналізованого МЕМС-пристрою, тому поставлена задача автоматизації процесу побудови графіків є актуальною.

Основна частина

У системі ANSYS для відображення вихідних даних про максимальні значення використовується постпроцесор POST1, натомість для побудови графіків використовують постпроцесор POST26. Постпроцесор POST26 дає змогу користувачу побудувати графічні залежності для вибраного скінченного елемента, максимальних напружень/деформацій за різних навантажень, температури, або для різних форм коливань. Для побудови графічних залежностей у системі ANSYS розроблений алгоритм [6], який зображено на рис. 1.