

4. Советов Б. Я., Яковлев С. А., Моделирование систем // М.: Высшая школа., 2001. – 343 с.
5. Введение в математическое моделирование/Учебное пособие. Под ред. П. В. Трусова // М.: Логос, 2004. – 294 с.
6. Поспелов Д. А. Ситуационное управление. Теория и практика // М.:Наука, 1999. – 288с.
7. Питерсон Дж. Теория сетей Петри и моделирование систем // М: Мир, 1984. – 264с.
8. Зайцев Д. А. Универсальная сеть Петри / Кибернетика и системный анализ. – № 4. –2012. – С. 24-39.
9. Мараховский В. Б., Розенблюм Л. Я., Яковлев А. В. Моделирование параллельных процессов. Сети Петри //СПб: Проф. литература, IT-Подготовка, 2014. – 400 с.
10. Ковбатьюк М. В., Ткаченко О.І., Міщенко В. В. Продукційно-ситуаційна модель розвитку транспортної галузі / Водний транспорт. – 2010. – Вип.11. – С.28-38.
11. Ткаченко О. И. Использование сетей Петри для ситуационного диалога в маршрутных системах ПРОЦЕСС // Средства представления знаний в информационных технологиях: Сб.научн.тр. – Киев: ИК АНУ, 1992. – С.33-38.

Ткаченко О.І., Ткаченко О.А., Ткаченко К.О.

МОДЕЛИРОВАНИЕ СИСТЕМЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ НА ОСНОВЕ СЕТЕЙ ПЕТРИ

В статье рассмотрены вопросы моделирования сложной экономической системы, которой является система подготовки специалистов. Предложено для использования модель, которой является сеть Петри. Формализованное описание модели позволяет определить состояние системы, проанализировать и спрогнозировать действия по ее развитию и совершенствованию, оказать поддержку принятия управленческих решений.

Ключевые слова: моделирования сложных систем, система подготовки специалистов, сеть Петри, управленческое решение по развитию и совершенствованию системы подготовки специалистов, состояние системы, параметры модели.

Tkachenko O.A., Tkachenko O.I., Tkachenko K.

MODELING OF SYSTEM OF TRAINING OF SPECIALISTS BASED ON PETRI NETS

The paper deals with the modeling of complex economic system, which is a system of training of specialists. It is proposed to use the model, which is a Petri net. A formal description of the model to determine the state of the system, to analyze and predict the actions to develop and improve the system of training, to support management decision-making.

Keywords: modeling of complex systems, system training, Petri nets, management solutions for the development and improvement of training, system state, parameters of the model.

УДК 004.031.2/.4/.42;004.384; 004.78;004.451.7;004.832.28; 005

Денєжко С.А.

МОДЕЛЬ ПОДІЄ-КЕРОВАНОЇ АРХІТЕКТУРИ ОБРОБКИ ПОДІЙ

У даній роботі представлена функціональна модель подіє-керованої архітектури (ПКА) обробки подій в програмі PNEditor, яка використовує мову мереж Петрі для опису моделей і має можливість швидкого конструювання алгоритмів імітації систем з великою кількістю подій. Концептуальна модель демонструє процес підготовки та обробки подій, які генеруються виробниками подій для використання споживачами подій. Формальні об'єкти

концептуальної моделі ПКА описані у вигляді умов і подій. Це забезпечує концептуальний погляд на архітектуру обробки подій та на ключові компоненти, необхідні для створення ефективних систем обробки подій.

Ключові слова: подіє-керована архітектура, подія, обробка подій, сервіс-орієнтована архітектура, мережа Петрі.

В основі створення подіє-керованої архітектури (ПКА) лежить уявлення про набір основних компонентів (фаз), які певним чином реагують на різні події. Ці події можуть бути як внутрішніми, так і зовнішніми. Зовнішніми є такі події, як натискання кнопки миші або надходження даних за допомогою мережевого з'єднання. Існує три основні стилі обробки подій: простий, потоковий і складний [4,7].

Для створення більш повної функціональної моделі в роботі ПКА виділимо три фази: 1 – фаза генерування подій; 2 – фаза обробки подій; 3 – фаза споживання подій.

Потік обробки не буває єдино можливим і стандартним, і між цими трьома фазами існує дуже обмежений взаємозв'язок. Тому ці три фази можуть виникати в різні періоди часу, і між генерованими подіями і спожитими подіями може бути тільки причинно-наслідковий взаємозв'язок. Компонентами концептуальної архітектури, залученими у фазу генерування, є головним чином компоненти генератора подій. Їх роль в генеруванні подій може бути абсолютно технічною, оскільки вони відповідають в основному за забезпечення технічних рівнів, але можуть бути орієнтовані і на бізнес-складову (тут може бути реалізована як певна бізнес-орієнтована логіка обробки подій, так і агенти обробки подій). У фазі обробки подій можуть брати участь різні потоки обробки із залученням різних компонентів в залежності від кількості оброблюваних подій. Одним з можливих основних варіантів обробки є модель публікація / підписка. При прийомі повідомлення про подію воно перехоплюється і розподіляється по різних каналах. Ці канали публікуються в реєстрі подій щоб бути доступними для споживачів подій або компонентів сервісів підписки. Споживачі подій приймають повідомлення про подію. Фаза споживання події відбувається в основному на рівні «Обробки подій», асоційованим зі споживачем подій. На цій фазі споживач подій приймає повідомлення про подію і обробляє його, покладаючись на свій локальний рівень обробника подій.

На рис. 1 показана спрощена модель подіє-керованої архітектури з фазами. Потік подій спрямований від виробника до споживача, в цьому ж порядку розташовані компоненти.

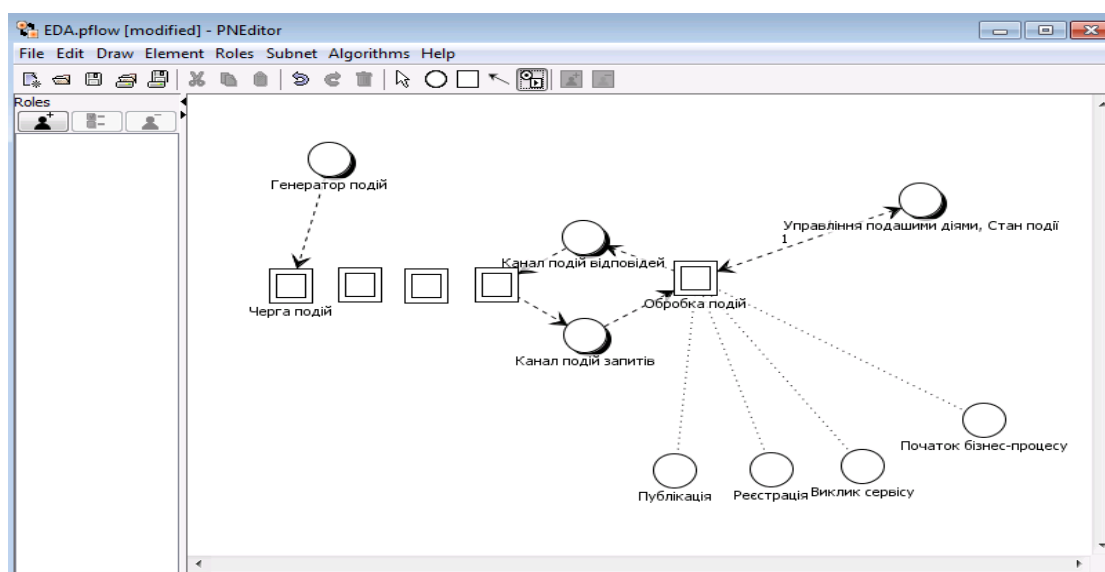


Рис.1. Спрощена модель ПКА з фазами

В якості моделюючого апарату було вирішено вибрати мережі Петрі [2]. Подія в мережі Петрі – це спрацювання переходу в мережі, при якому мітки з вхідних позицій цього

переходу переміщуються у вихідні позиції. Події відбуваються миттєво або різночасно при виконанні деяких умов. Для того, щоб створити нову структуру управління подіє-керованої архітектури, можна скористатися програмою PNEditor [6], яка являє собою спеціальну моделюючу систему, використовуючи мову мереж Петрі для опису моделей (рис.2). PNEditor поряд з графічною побудовою мережі Петрі дозволяє створювати підкласи в переходах, а також зберігає отриману мережу на мові інтернет-розмітки PNML (Petri Net Markup Language – мова розмітки мережі Петрі), яка є поширеною мовою розмітки XML (eXtensible Markup Language – розширювана мова розмітки) з можливістю подальшого використання та аналізу [6,7].

Візуально зручним є графічне представлення мережі Петрі. Якщо ми позначимо колом події, а квадратом – переходи (умови), в якості відображень переходів приймемо орієнтовані стрілки, то отримаємо граф мережі Петрі.

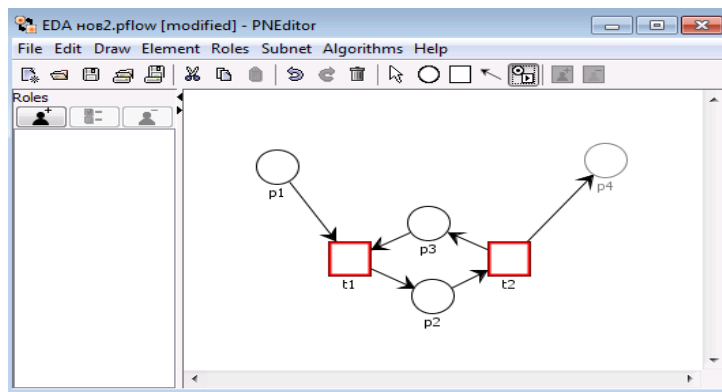


Рис.2. ПКА простої обробки подій на мультіграфі мережі Петрі

Послідовність простої (класичної) обробки подій (рис.2) виглядає наступним чином. Спочатку подія SOA поступає у генератор подій, генератор подій передає подію у канал подій, далі канал подій запитів приймає подію та передає подію в обробку подій, далі подія передається у канал відповідей і знову поступає у канал подій запитів, канал подій запитів приймає подію та знову передає її в обробку – останній крок в простій обробці подій управління цією подією. Даний опис можна подати у вигляді таблиці:

Подія: P	Умова: T
1 – режим очікування події SOA у генераторі подій	1 – передача події у канал подій запитів
2 – канал подій запитів приймає подію	2 – передача події в обробку
3 – передача події у канал відповідей	1 – передача події у канал подій запитів
2 – канал подій запитів приймає подію	2 – передача події в обробку
4 – управління подальшими діями (стан події)	

Граф G мережі Петрі – це двочастковий орієнтований мультіграф,

$$G = (V, A),$$

де $V = P \cup T$ – безліч вершин, $A = \{a_1, a_2, \dots, a_r\}$, – комплект спрямованих дуг, $a_i = (v_j, v_k)$, де $v_j, v_k \in V$; і для будь спрямованої дуги або $v_j \in T, v_k \in P$, або $v_j \in P, v_k \in T$.

Доведення еквівалентності уявлень можна знайти в [2], а на рис.2 наведено EDA простої обробки подій в графічному поданні.

Мережа Петрі C є четвіркою,

$$C = \{P, T, I, O\}$$

де $P = \{p_1, p_2, \dots, p_n\}$ – безліч позицій, $n > 0$, $T = \{t_1, t_2, \dots, t_m\}$ – безліч переходів, $m > 0$, таких, що $P \cap T = \emptyset$. $I: T \rightarrow P^\infty$ – є вхідною функцією-відображенням з переходів у комплекти позицій. $O: T \rightarrow P^\infty$ – вихідна функція-відображення з переходів у комплекти позицій [3].

$$C = (P, T, I, O),$$

$$P = \{p_1, p_2, p_3, p_4\},$$

$$T = \{t_1, t_2\},$$

$$I(t_1) = \{p_1, p_3\}, \quad O(t_1) = \{p_2\},$$

$$I(t_2) = \{p_2\}, \quad O(t_2) = \{p_3, p_4\}$$

Розглянемо процес обробки подій та процес управління подіями більш детально (рис. 3), тому що вони є не такими і простими, як показано на рис.2.

Процес обробки поділяється на кілька етапів: захоплення події, бізнес-процес, виклик сервісу та реєстрація події.

Процес управління подіями поділяється на прийняття події: людиною або панелью інструментів, або подія призначена для бізнес-процесу, або для бази даних чи для агента.

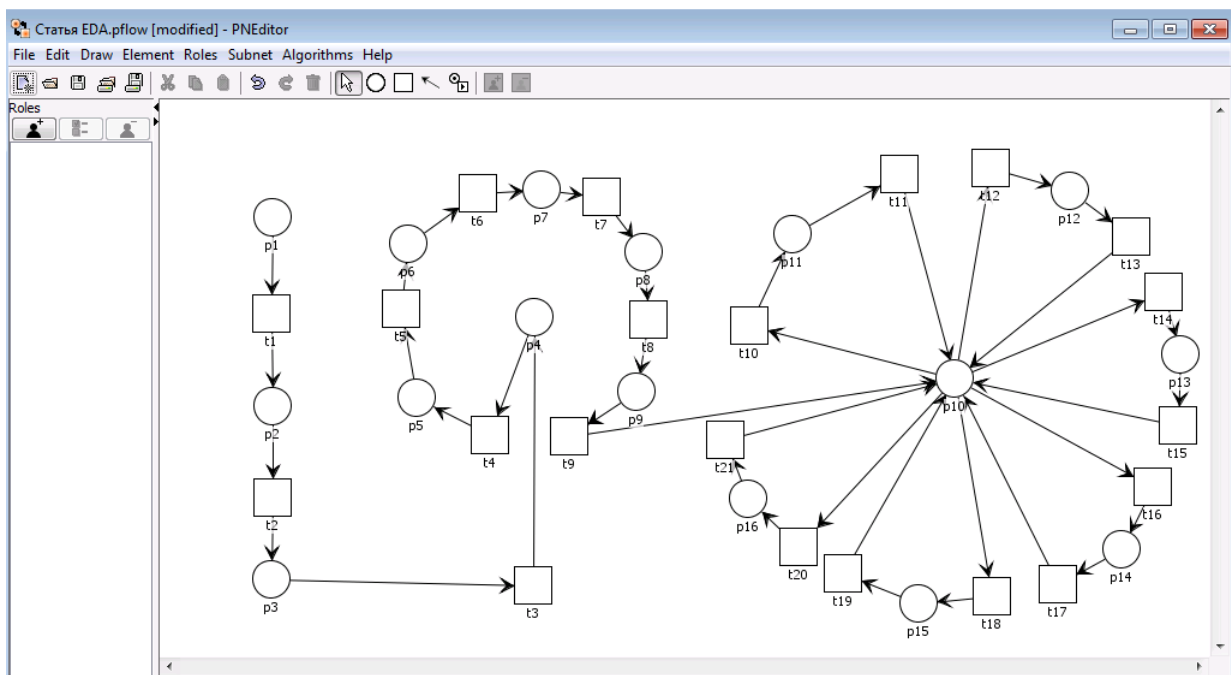


Рис. 3. ПКА простої обробки подій на мультіграфі мережі Петрі з управлінням опублікованими подіями

Послідовність простої обробки подій тепер виглядає більш розширеною (рис. 3):

Подія: P	Умова: T
1 – режим очікування події SOA у генераторі подій	1 – передача події у канал подій запитів
2 – канал подій запитів приймає подію	2 – передача події у процесор подій
3 – процесор подій приймає подію	3 – передача події в обробку
4 – обробка події	4 – передача обробленої події
5 – захоплення події	5 – захоплена подія передається у БП
6 – початок бізнес-процесу	6 – передача події на оцінювання
7 – оцінка події у бізнес-процесі	7 – оцінена подія передається сервісу
8 – виклик сервісу	8 – сервіс передає подію на реєстрацію
9 – реєстрація події	9 – зареєстрована подія передається на публікацію
10 – публікація події	10 – опублікована подія передається людині
11 – якщо подія створена тільки для людини, вона залишається	11 – передача події
12 – якщо подія створена тільки для ПІ, вона залишається	12 – опублікована подія передається у ПІ
	13 – передача події
13 – якщо подія створена тільки для БП, вона залишається	14 – опублікована подія передається у активний БП
	15 – передача події
14 – якщо подія створена тільки для БД, вона залишається	16 – опублікована подія передається у БД
	17 – передача події
15 – якщо подія створена тільки для агента, вона залишається	18 – опублікована подія передається агенту
	19 – передача події
16 – додатки	20 – опублікована подія приймається
	21 – передача події

На рис.3 розглянуто просту обробку подій. Тепер розглянемо обробку потоку подій, яка використовується для управління потоком інформації в реальних системах (рис. 4), коли у процес обробки подій може поступити подія не з нашого SOA, а подія від інших розробників. Якщо це станеться, тоді обробити подію у простій (класичній) обробці подій буде неможливо.

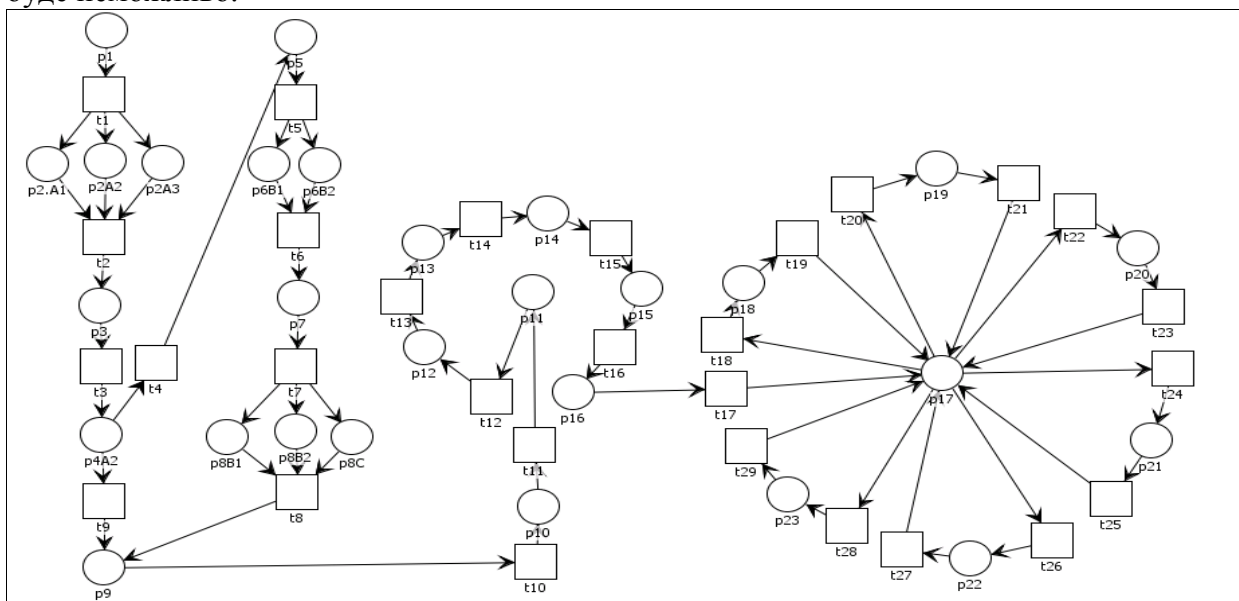


Рис. 4. ПКА обробки потоку подій на мультіграфі мережі Петрі

Алгоритм роботи програми, заснованої на подіє-керованій моделі, наступний:

Подія: P	Умова: T
1 – режим очікування події SOA	1 – передача події
2 – подія поступила	2 – передача події у локальний фільтр
3 – локальний фільтр приймає подію	3 – сортування подій
4 – подія згенерована	4 – передача незгенерованої події SOA
5 – ресурс, генеруючий інші події, приймає подію	5 – згенерована подія передається у канал подій
6 – подія поступила	6 – подія передається у router подій
7 – router приймає подію	7 – сортування подій інших розробників
8 – події згенеровані	8 – згенеровані події інших розробників передаються у канал подій
	9 – згенеровані події SOA передаються у канал подій
9 – канал подій приймає події	10 – канал подій передає події у процесор подій
10 – процесор подій приймає події	11 – процесор подій передає події в обробку
11 – обробка події	12 – передача події
12 – захоплення події	13 – передача події у бізнес-процес
13 – початок БП	14 – передача події на оцінювання
14 – оцінка події у БП	15 – подія передається сервісу
15 – виклик сервісу	16 – передача події на реєстрації
16 – реєстрація події	17 – передача зареєстрованої події на публікацію
17 – публікація події	18 – опублікована подія передається людині
18 – якщо подія створена тільки для людини, вона залишається	19 – передача події, якщо вона непотрібна або якщо треба скористатися нею ще
	20 – опублікована подія передається у панель інструментів
19 – якщо подія створена тільки для ПП, вона залишається	21 – передача події
20 – якщо подія створена тільки для БП, вона залишається	22 – опублікована подія передається у активний БП
	23 – передача події
21 – якщо подія створена тільки для БД, вона залишається	24 – опублікована подія передається у БД
22 – якщо подія створена тільки для агента, вона залишається	25 – передача події
	26 – опублікована подія передається агенту
23 – додатки	27 – передача події
	28 – опублікована подія приймається
	29 – передача події

Висновки. Стаття розкриває поняття подіє-керованої архітектури. Як дослідницьке завдання, автором була визначена спроба змоделювати просту (класичну) обробку процесу подій у ПКА, а також обробку потоку подій від реальних систем у ПКА. У роботі представлена функціональна модель ПКА обробки подій, що складається з концептуальної архітектури, заснованої на концепціях (фазах) мережі обробки подій у програмі PNEditor. Ця концептуальна архітектура демонструє процес підготовки та обробки події, що генерується виробниками подій для використання споживачами подій; включає можливості, які можуть знадобитися стороні виробника або споживача, наприклад, можливості деякої простої обробки подій або потоку подій. Призначення цієї концептуальної архітектури полягає в ідентифікації всіх компонентів, які можуть знадобитися при реалізації системи обробки подій, але для конкретної реалізації або сценарію швидше за все будуть потрібні тільки обрані компоненти. Представлено кілька сценаріїв, які демонструють використання подіє-керованої архітектури обробки подій для отримання результату в залежності від потоку подій. Запропоновано функціональну модель реальної подіє-керованої архітектури та побудовано її формальну модель у мережі Петрі. Однак, дана задача потребує подальших досліджень.

ЛІТЕРАТУРА

1. Денежко С. А. Подходы к методологии EDA. [Электронный ресурс] – Режим доступа: <http://interactive-plus.ru/e-articles/conf-6/conf-6-2148.pdf>
2. Денежко С. А, Полиновский В. В. Предприятие реального времени с использованием принципов EDA и SOA. Сборник научных статей по итогам Международной научно-

практической конференции, 26-27 сентября 2013 года, г. Санкт-Петербург. – СПб.: Изд-во «КультИнформПресс», 2013 – С. 148-151.

3. Сети Петри, описание системы [Электронный ресурс] – Режим доступа: http://www.iacr.dvo.ru/lab_11/otchet/ot2000/pn3.html
4. Event-Driven Architecture Overview. [Электронный ресурс] Event-Driven SOA Is Just Part of the EDA Story By Brenda M. Michelson Sr. VP and Sr. Consultant, Patricia Seybold Group. – Режим доступа: <http://www.omg.org/soa/Uploaded%20Docs/EDA/bda2-2-06cc.pdf>
5. Koschmider Agnes, Oberweis Andreas. Ontology Based Business Process Description. Proceedings of the CAiSE'05 WORKSHOPS.– MIT Press, 2005. – Porto, 2005. – pp. 321-333.
6. PNEditor [Электронный ресурс] – Режим доступа: <http://www.pneditor.org/>
7. Wikipedia The Free Encyclopedia [Электронный ресурс] / Event-driven architecture – Режим доступа: http://en.wikipedia.org/wiki/Event-driven_architecture

Денежко С.А.

МОДЕЛЬ СОБЫТИЙНО-УПРАВЛЯЕМОЙ АРХИТЕКТУРЫ ОБРАБОТКИ СОБЫТИЙ

В данной работе представлена функциональная модель событийно-управляемой архитектуры обработки событий в программе PNEditor, которая использует язык сетей Петри для описания моделей и обладает возможностью быстрого конструирования алгоритмов имитации систем с большим количеством событий. Концептуальная модель демонстрирует процесс подготовки и обработки событий, которые генерируются производителями событий для использования потребителями событий. Формальные объекты концептуальной модели событийно-управляемой архитектуры описаны в виде условий и событий. Это обеспечивает концептуальный взгляд на архитектуру обработки событий и на ключевые компоненты, необходимые для создания эффективных систем обработки событий.

Ключевые слова: *событийно-управляемая архитектура, событие, обработка событий, сервис-ориентованная архитектура, сеть Петри.*

Denezhko S.

MODEL EVENT DRIVEN ARCHTECTURE PROCESS DATA

The article presents the functional model of controlled architecture of processing data in program PNEditor, which uses the language of network Petri for describing models and the possibility of rapidly constructing algorithm of imitation system with a lot of events. Conceptual model displays the process of preparation and performance events/data, which are generated by producers of events for using consumer data. Formal objects of conceptual model look like conditions and events. It provides the conceptual view to architecture of processing data and key components, which are necessary for creating effective system of processing data.

Keywords: *controlled architecture, event, preparation of event, service oriented architecture, Petri network.*