

## ИНСТРУМЕНТАРИЙ СИНТЕЗА МОДЕЛЕЙ ЖИЗНЕННЫХ ЦИКЛОВ РАЗВИВАЮЩИХСЯ ОБЪЕКТОВ

А. В. Ермаков

Московский инженерно-физический институт

*Рассмотрены основные возможности системы ER&LC 2.0: ER-моделирование сущность—связь; возможность построения динамической модели системы в виде общей сети Петри (СП) на основе синтеза жизненного цикла (ЖЦ) сущностей, моделируемых конечными автоматами; программный модуль преобразования конечного автомата в СП, моделирующую ЖЦ каждой сущности; процедура и программный модуль синтеза частных СП в общую сеть; механизм генерации серверного компонента, контролирующего динамические процессы в информационных системах на уровне событий и основанного на общей СП. В разработанной системе жизненные циклы сущностей (бизнес-логика) моделируются общей СП. Фактически формируется надстройка над реляционной моделью, промежуточный уровень логики, контролирующий семантическую целостность данных и операций над ними.*

CASE (Computer Aided Software/System Engineering)-технология представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения информационных систем (ИС), поддержанную комплексом средств автоматизации [1]. CASE-технология позволяет максимально систематизировать и автоматизировать все этапы создания ИС. Ключевыми условиями решения задачи сокращения сроков и затрат при создании ИС являются:

- наличие средств моделирования и анализа создаваемых систем, что позволяет избежать критических ошибок на ранних этапах проекта;
- наличие средств автоматической генерации кода на основе модели.

Хотя CASE-средства проектирования ИС, основанные на реляционной модели данных, и позволяют осуществлять автоматическую гене-

рацию клиентского приложения, но данная модель не содержит информации о бизнес-процессах. Как следствие — сгенерированный код не может обеспечить функциональность приложения со сложной бизнес-логикой. Для решения данной проблемы необходимо расширение реляционной модели.

Сокращение сроков и стоимости создания ИС со сложной бизнес-логикой определяет высокую актуальность работ по созданию CASE-технологии автоматизированного проектирования ИС на основе расширенной реляционной модели.

Расширение реляционной модели данных, основанное на представлении каждой сущности предметной области как развивающегося объекта (РО), моделирование ЖЦ РО в виде конечных автоматов, анализ ИС с использованием аппарата СП, программная реализация описы-



ваемых моделей в CASE-системе ER&LC 2.0 System [2] — тема данной статьи.

### Модели жизненных циклов развивающегося объекта

Под развивающимся объектом обычно понимают объект, рассматриваемый в течение всего периода его существования, который получил название "жизненный цикл" и включает возникновение, функционирование и исчезновение объекта. Понятие РО является более широким по отношению к понятию "динамический объект", в котором рассматривается лишь функционирующий объект без его возникновения и исчезновения [3].

В системном анализе принято выделять две основные процедуры: структуризация объекта и описание компонент структурированного объекта [4].

### Конечно-автоматная модель

Конечный автомат позволяет моделировать поведение отдельного объекта (экземпляра сущности). Он описывает поведение в терминах последовательности состояний, через которые проходит объект в течение своей жизни, отвечая на события, а также его реакцию на эти события.

Автоматы используются для моделирования динамических аспектов отдельных сущностей. Когда происходит событие, в зависимости от текущего состояния объекта имеет место та или иная деятельность. Хорошо структурированные автоматы подобны хорошо структурированным алгоритмам — они эффективны, адаптируемы к разным ситуациям и просты для понимания.

Итак, конечный автомат представляет собой шестерку:

$$A = (Q, \Sigma, \Delta, \delta, S, F),$$

где  $Q$  — множество всех возможных состояний сущности;

$\Sigma$  — множество всех возможных событий (входной алфавит), которые могут произойти с сущностью;

$\Delta$  — множество всех возможных действий (выходной алфавит), которые могут произойти;

$\delta$  — функция перехода сущности в следующее состояние (определена на Декартовом произведении  $Q \times \Sigma$ );

$S$  (принадлежит  $Q$ ) — начальное состояние;

$F$  (принадлежит  $Q$ ) — множество конечных состояний [5].

### Модель с использованием аппарата сетей Петри

Сети Петри — инструмент моделирования систем. Теория СП делает возможным модели-

рование системы математическим представлением ее в виде СП. Первоначально СП использовались для описания взаимодействующих автоматов, но их аппарат оказался очень удобным для решения задач, связанных с анализом, моделированием и представлением причинно-следственных связей в сложных системах взаимодействующих объектов, в частности, для анализа, моделирования и проектирования ИС.

Сеть Петри состоит из четырех элементов: множество *позиций*  $P$ , множество *переходов*  $T$ , *входная* функция  $I$ , *выходная* функция  $O$ . Входная и выходная функции связаны с переходами и позициями. Входная функция  $I$  отображает переход  $t_j$  в множество позиций  $I(t_j)$ , называемых входными позициями перехода. Выходная функция  $O$  отображает переход  $t_j$  в множество позиций  $O(t_j)$ , называемых выходными позициями перехода. Структура СП определяется ее позициями, переходами, входной и выходной функциями [5, 6].

Сеть Петри является четверкой:

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

где  $P = \{p_1, p_2, \dots, p_n\}$  — конечное множество позиций;

$$n \geq 0;$$

$T = \{t_1, t_2, \dots, t_m\}$  — конечное множество переходов;

$$m \geq 0.$$

Множество позиций и множество переходов не пересекаются;

$I$  — входная функция, отображение из переходов в комплекты позиций;

$O$  — выходная функция, отображение из переходов в комплекты позиций.

Системы, моделируемые СП, могут быть представлены следующим образом:

$$S = \langle C_1, C_2, \dots, C_i, C_n \rangle, \quad (1)$$

где  $C_i$  —  $i$ -я компонента системы.

Каждая компонента сама может быть системой, но ее поведение можно описать независимо от других компонент системы, за исключением точно определенных взаимодействий с другими компонентами.

### Проектирование динамической модели ИС

В качестве модели ИС будем использовать (1), тогда компонентами ИС являются сущности, состояния которых представляются их ЖЦ, а работа ИС моделируется исполнением соответствующей СП.

Общая СП, которая будет описана ниже, представляет собой композицию ЖЦ отдельных сущностей, связанных общими входными и выходными позициями. Использование этой модели дает возможность на событийном уровне моделировать совместное развитие экземпляров различных сущностей.

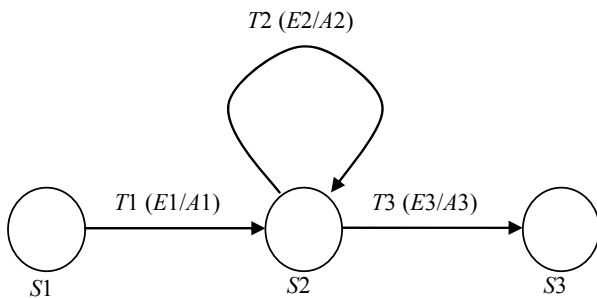
Кроме того, общая СП используется в качестве шаблона ядром серверного блока ИС.

*Процедура преобразования конечного  
автомата в сеть Петри.  
Получение частной сети Петри*

Для конечного автомата  $(Q, \Sigma, \Delta, \delta, S, F)$  определяем СП  $(P, T, I, O)$ , где  $P = \{E \cup A \cup S\}$ , таким образом, что:

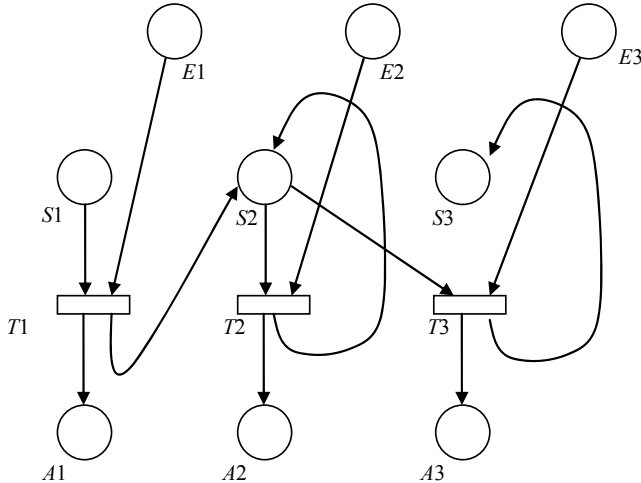
$$\begin{aligned} E &= \Sigma; \\ A &= \Delta; \\ S &= Q; \\ T &= \{t_{q,\sigma} \mid q \in Q \text{ и } \sigma \in \Sigma\}; \\ I(t_{q,\sigma}) &= \{q, \sigma\}; \\ O(t_{q,\sigma}) &= \{\delta(q, \sigma)\}. \end{aligned}$$

На рис. 1 и 2 представлены автомат и соответствующая ему частная СП.



**Рис. 1. Конечный автомат:**

$S1, S2, S3$  — состояния;  $T1, T2, T3$  — переходы;  $E1, E2, E3$  — входные события;  $A1, A2, A3$  — выходные события



**Рис. 2. Частная сеть Петри, соответствующая автомату на рис. 1**

(Обозначения те же, что на рис. 1)

Таким образом, в частной СП существуют три подмножества позиций, соответствующих множествам состояний, входному и выходному алфавиту автомата.

*Процедура композиции частных сетей.  
Получение общей сети Петри*

Рассмотрим процедуру композиции (синтеза) частных сетей, получение которых из автомат-

ных моделей описано выше, в общую СП. Нужно отметить, что существование общей сети следует из самой процедуры построения, которая будет рассмотрена далее. Доказать неповторимость можно лишь для этой конкретной процедуры построения, так как в общем случае она не является единственно возможной. Иными словами, предлагаемый механизм композиции является одним из множества возможных и дает в данном случае единственное решение.

Формально композиция  $N$  частных сетей  $(P_1, T_1, I_1, O_1), \dots, (P_N, T_N, I_N, O_N)$ , где  $P_i = \{E_i, A_i, S_i\}$ ,  $\exists i, j : E_i \cap E_j \neq \emptyset, A_i \cap A_j \neq \emptyset, \forall i, j : S_i \cap S_j = \emptyset$ , представляет собой сеть Петри  $(P_0, T_0, I_0, O_0)$ , где  $P_0 = \{E_0, A_0, S_0\}$ ;

$$E_0 = \cup E_j = \{e_1, \dots, e_K\} \forall i, j \ e_i \neq e_j;$$

$$A_0 = \cup A_j = \{a_1, \dots, a_L\} \forall i, j \ a_i \neq a_j;$$

$$S_0 = \cup S_j = \{s_1, \dots, s_M\} \forall i, j \ s_i \neq s_j;$$

$$T_0 = \{t_{01}, \dots, t_{0K}\};$$

$$I_0(t_{0i}) = \cup I(t_j), \text{ где } t_{0i} = \cup t_j \text{ и } I(t_j) = \{e_j\};$$

$$O_0(t_{0i}) = \cup O(t_j), \text{ где } t_{0i} = \cup t_j \text{ и } I(t_j) = \{e_j\}.$$

Рассмотрим композицию на примере синтеза двух СП.

Первая сеть (рис. 3):

$$E = \{E1, E2, E3\};$$

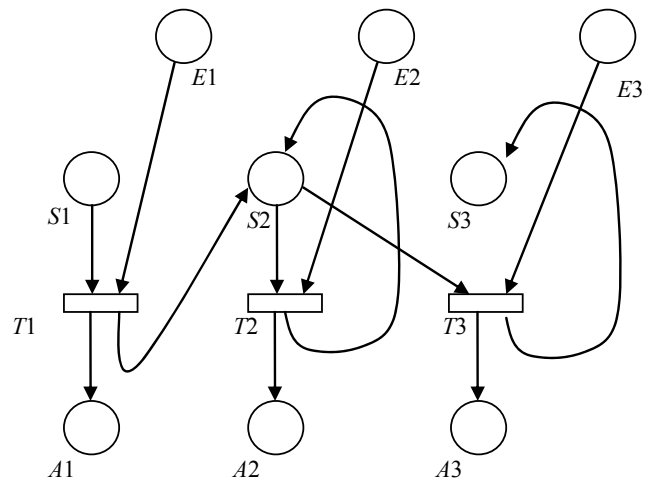
$$A = \{A1, A2, A3\};$$

$$S = \{S1, S2, S3\};$$

$$T = \{T1, T2, T3\};$$

$$I(t_1) = \{S1, E1\}, I(t_2) = \{S2, E2\}, I(t_3) = \{S2, E3\};$$

$$O(t_1) = \{A1, S2\}, O(t_2) = \{A2, S2\}, O(t_3) = \{A3, S3\}.$$



**Рис. 3. Первая частная сеть Петри**  
(Обозначения те же, что на рис. 1)

Вторая СП (рис. 4):

$$E = \{E1, E3, E4\};$$

$$A = \{A3, A4, A5\};$$

$$S = \{S4, S5, S6\};$$

$$T = \{T4, T5, T6\};$$

$$I(t_4) = \{S4, E1\}, I(t_5) = \{S5, E4\}, I(t_6) = \{S6, E3\};$$

$$O(t_4) = \{A4, S5\}, O(t_5) = \{A5, S5\}, O(t_6) = \{A3, S6\}.$$

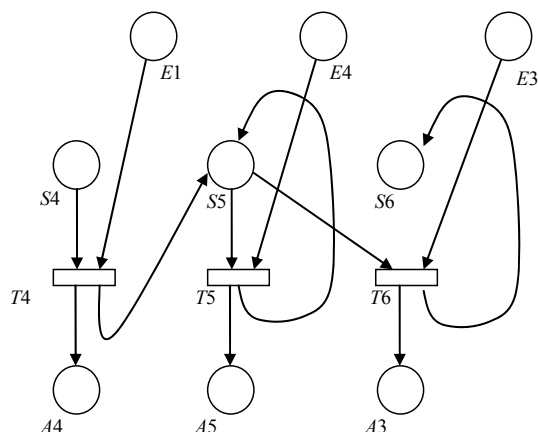


Рис. 4. Вторая частная сеть Петри  
(Обозначения те же, что на рис. 1)

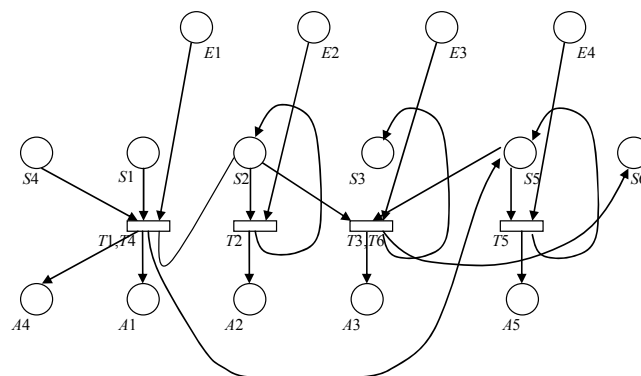


Рис. 5. Общая сеть Петри  
(Обозначения те же, что на рис. 1)

Результатом композиции двух частных сетей служит общая СП (рис. 5):

$E_0 = \{E1, E2, E3, E4\};$   
 $A_0 = \{A1, A2, A3, A4, A5\};$   
 $S_0 = \{S1, S2, S3, S4, S5, S6\};$   
 $T_0 = \{T1+T4, T2, T3+T6, T5\};$   
 $I_0(t_1+t_4) = \{S1, S4, E1\}, I_0(t_2) = \{S2, E2\},$   
 $I_0(t_3+t_6) = \{S2, S5, E3\}, I_0(t_5) = \{S5, E4\};$   
 $O_0(t_1+t_4) = \{A1, A4, S2, S5\}, O_0(t_2) = \{A2, S2\},$   
 $O_0(t_3+t_6) = \{A3, S3, S6\}, O_0(t_5) = \{A5, S5\}.$

## CASE-система ER&LC 2.0

### Архитектура CASE-системы

Для описания архитектуры системы используется нотация UML [7]. На рис. 6 представлены основные варианты использования системы.

На рис. 7 приведена общая архитектура системы.

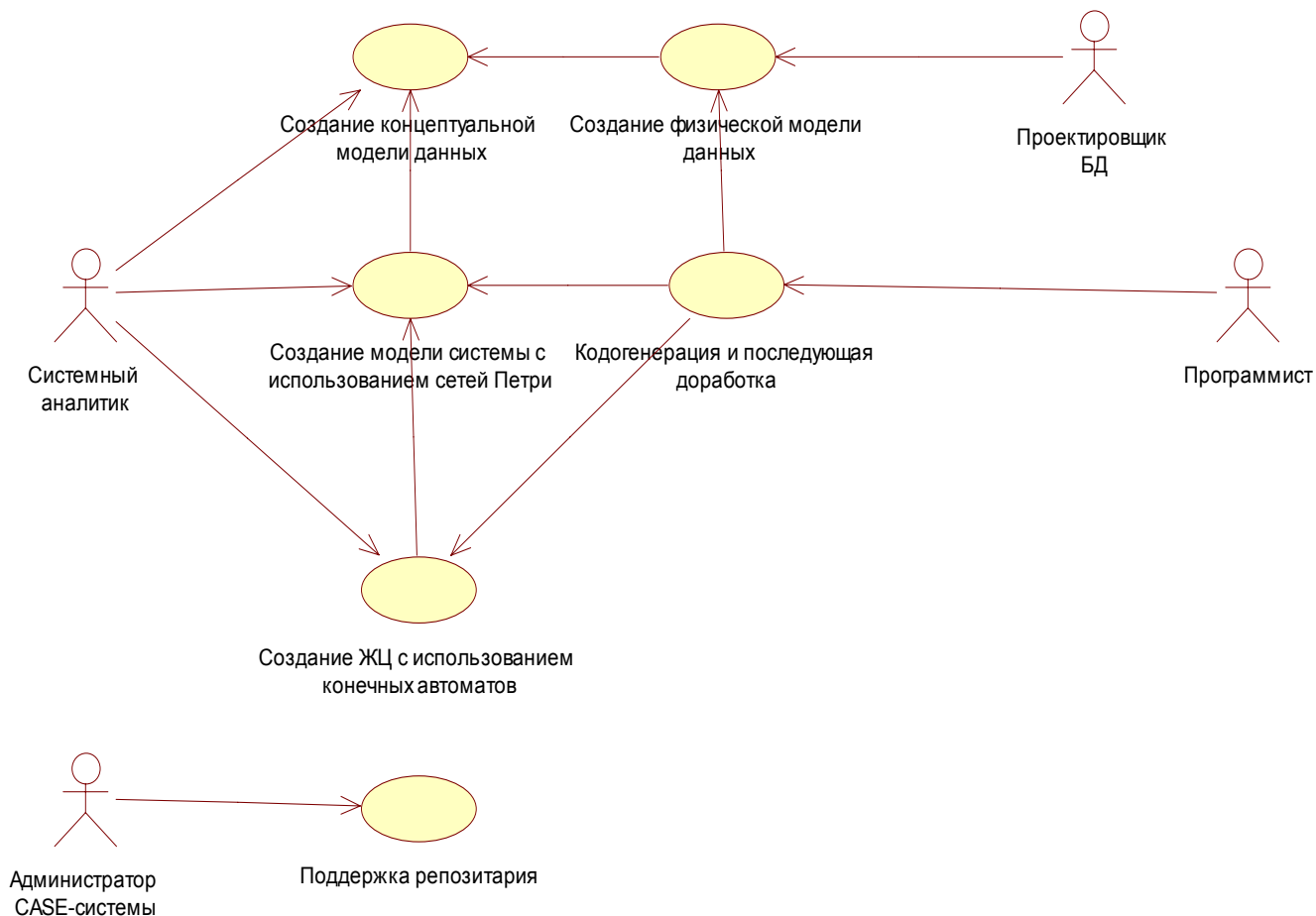


Рис. 6. Основные варианты использования системы

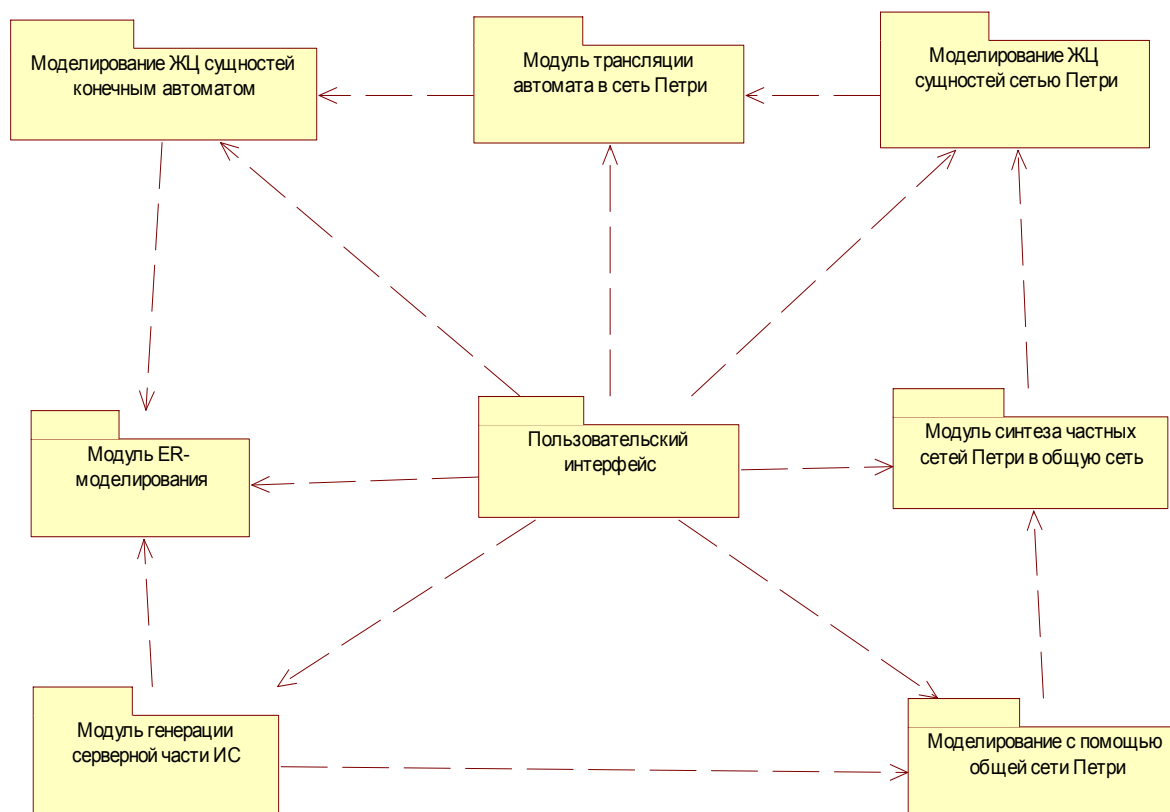


Рис. 7. Общая архитектура системы

### Информационные системы, создаваемые с помощью ER&LC System 2.0

Для создания серверной и клиентской частей ИС в ER&LC System 2.0 предусмотрен специальный модуль. Основная задача данного модуля — генерация программного кода серверной и клиентской частей — основных частей, составляющих проектируемую трехуровневую ИС.

Именно промежуточный слой (сервер) служит идеальным местом размещения логики, моделируемой CASE-средством.

Клиентская часть направляет запросы серверу и получает ответный код сервера. Запрос выглядит следующим образом: код сущности, код экземпляра сущности, событие (и клиент, и сервер оперируют общими кодами всех объектов, которые хранятся в служебных таблицах в БД, доступ к которым имеет только серверная часть). Сервер, приняв запрос от клиента, строит маркировку для той общей сети, на которой он собственно и построен. Далее сервер запускает сеть, чтобы определить дальнейшее развитие объектов. В результате экземпляры сущностей переходят из одних состояний в другие, фишки попадают в позиции, соответствующие выходным символам, т. е. действиям, которые запускают определенные действия с БД, объектами системы (в соответствии с бизнес-логикой предметной области) и т. д. Когда выполнение останавливается, состояния запоминаются в БД и система переходит в режим ожидания следующего клиентского запроса.

Различают два типа вызовов, направляемых от клиента к серверу:

код события — в этом случае сервер строит маркировку общей СП по *правилу 1*;

код сущности, код экземпляра сущности, код события — в этом случае сервер строит маркировку общей СП по *правилу 2*.

#### Правило 1:

- фишка помещается в позицию СП, соответствующую наступившему событию;
- фишки помещаются в позиции СП, соответствующие состояниям сущностей. Достаточно, чтобы существовал хотя бы один экземпляр сущности в данном состоянии;
- все остальные позиции очищаются.

#### Правило 2:

- фишка помещается в позицию СП, соответствующую наступившему событию;
- фишка помещается в позицию СП, соответствующую состоянию конкретного (передаваемого) экземпляра сущности;
- все остальные позиции очищаются.

Механизм генерации серверной части состоит из следующих шагов:

- построение общей СП моделируемой предметной области;
- инициализация глобальных параметров общей сети будущей серверной части (позиции, переходы и т. д.);
- генерация программного кода (модулей), содержащих реализацию основных серверных подсистем.

Последний этап происходит на основании ER-модели (создание подсистемы работы с БД), а также общей СП. Для каждой сущности создается необходимый набор компонентов, соответствующий подсети общей сети.

Рассмотрим более подробно процесс моделирования и получения частных СП:

1. Пусть имеются две сущности, связанные отношением 1:N. Тогда, используя функциональность ER и ЖЦ моделирования, получаем простейшую ER-модель системы и два конечных автомата, моделирующих ЖЦ (для каждой сущности). На рис. 8, 9 изображены ЖЦ этих сущностей.

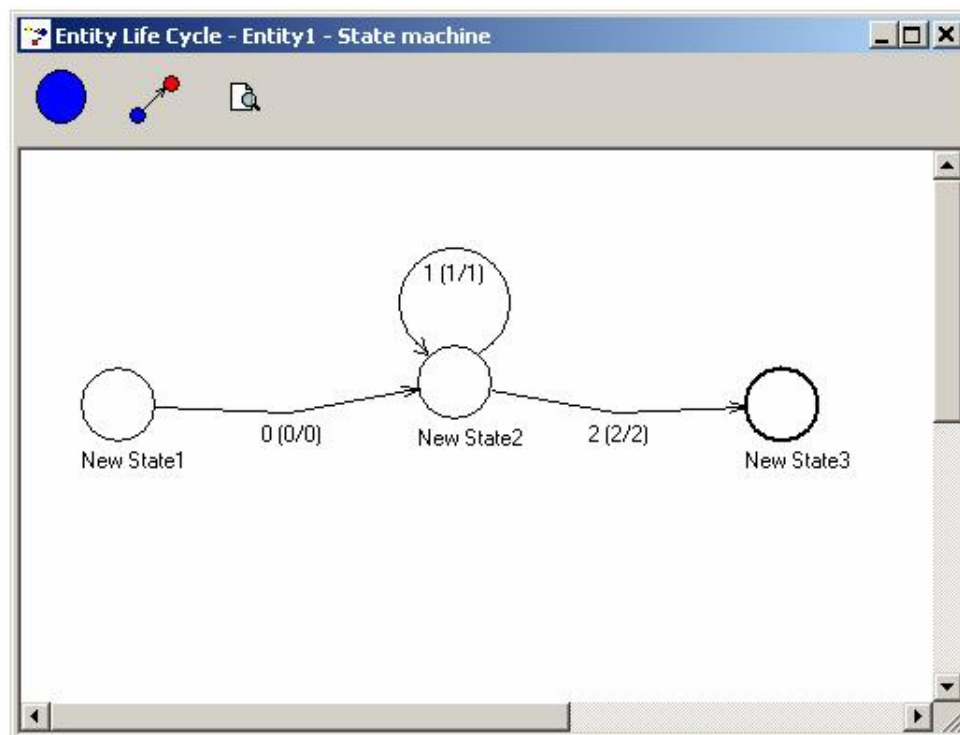


Рис. 8. ЖЦ первой сущности

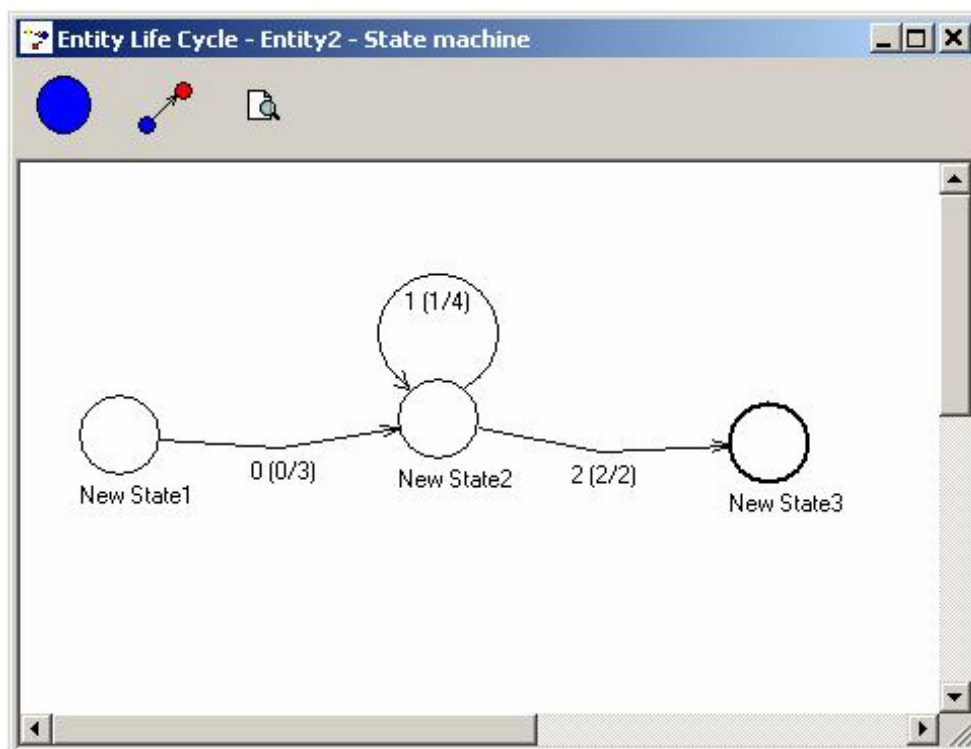


Рис. 9. ЖЦ второй сущности



Необходимо отметить, что входной и выходной алфавиты автоматов принадлежат общим множествам входных и выходных событий (для всех сущностей), т. е. каждое из пространств (событий и действий, соответственно) является общим для всей моделируемой области.

2. Далее с помощью механизма преобразования конечного автомата в СП получаем две сети (по одной для каждой сущности). Данное преобразование происходит автоматически на основании конечно-автоматных моделей. Переход от конечного автомата к СП основан на следующем преобразовании:

- все состояния автомата преобразуются в позиции СП (с типом "состояние");

- все переходы автомата преобразуются в переходы сети;

- все события и действия (входной и выходной алфавиты) преобразуются в позиции сети (с типом "событие" или "действие", соответственно);

- далее происходит связывание позиций и переходов сети на основании конечно-автоматных моделей (состояний, переходов, событий и действий) и преобразование данной информации во входную и выходную функции сети.

На рис. 10 изображены СП, соответствующие автоматам на рис. 8, 9.

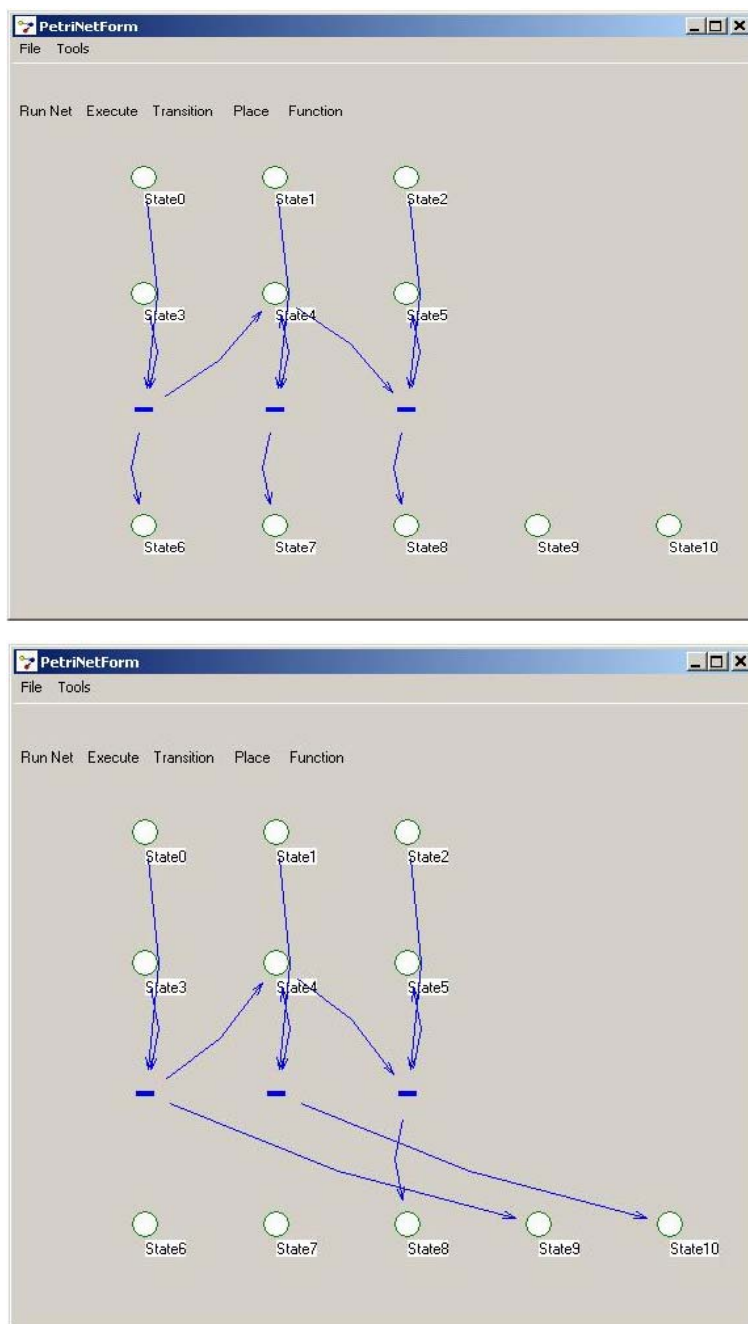


Рис. 10. Сети Петри для каждой сущности



После преобразования конечных автоматов в СП для каждой сущности, пользователь имеет возможность осуществлять моделирование уже с помощью этой сети, т. е. добавлять, удалять позиции и переходы, менять входные и выходные функции сетей, осуществлять стоимостной анализ каждой из сетей и т. д. В принципе частная сеть, полученная для сущности, не расширяет моделируемой области по сравнению с автоматом, но позволяет использовать для этого другой математический аппарат. Основное назначение частных сетей заключается в том, что они служат основой, компонентами синтеза общей СП.

Следующий шаг — генерация общей СП для обеих сущностей. Этот процесс основан на уже полученных сетях, моделирующих ЖЦ каждой сущности в отдельности. При активизации пользователем функции создания общей сети происходит автоматическое объединение всех сетей в одну по следующему правилу: общая сеть образуется из частных сетей путем их совмещения по позициям, соответствующим общим входным и

выходным символам (событиям или действиям); так как каждому автоматному переходу соответствует единственное событие (автомат детерминированный), то в случае общих входных событий объединяются и соответствующие им переходы частных СП. Другими словами, фактически, каждая сущность по-прежнему имеет свой ЖЦ в рамках общей сети, но все они объединены в одну большую сеть и синхронизированы на общие события и действия. При выполнении общей сети такой системы несколько экземпляров разных сущностей могут развиваться параллельно и независимо друг от друга. Общая сеть, соответствующая уже рассмотренным выше сетям, представлена на рис. 11.

Для преобразования подобной сети в информационную систему, отслеживающую поведение и развитие объектов, необходимо, чтобы каждая позиция общей сети имела информацию о том, к какой сущности (сущностям) она относится, а также, какому элементу конечного автомата позиция соответствует (состояние, событие, действие).

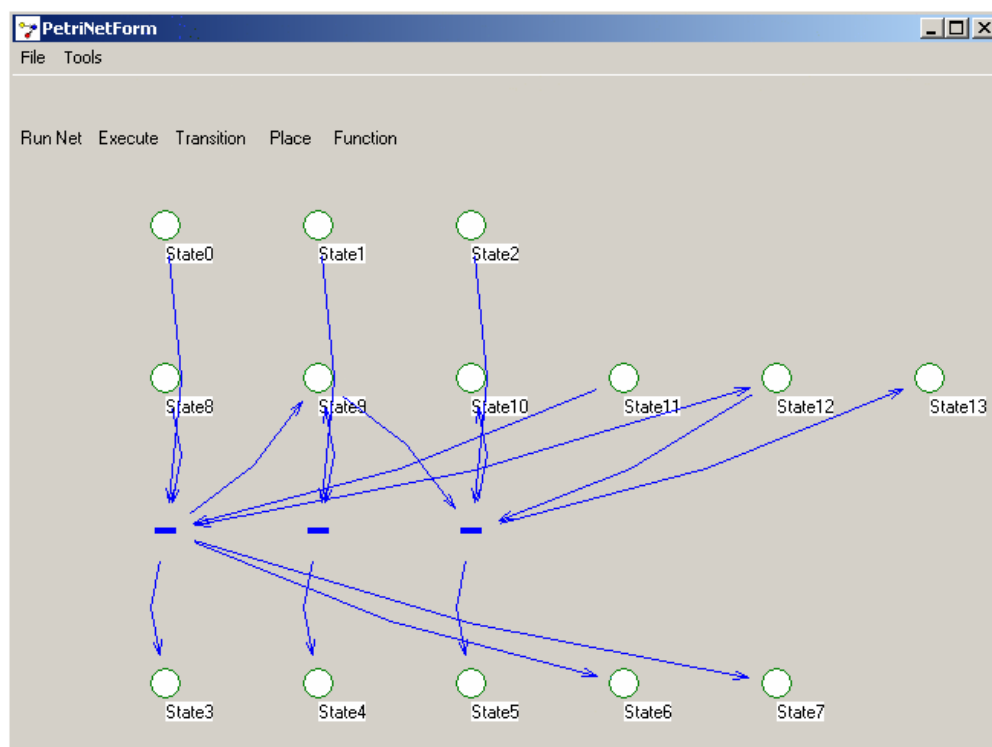


Рис. 11. Общая сеть Петри

#### Литература

1. Калянов Г. Н. CASE. Структурный системный анализ (автоматизация и применение). — М.: Лори, 1996.
2. Ермаков А. В. Реализация технологии CORBA при разработке ИС с использованием CASE-системы ER&LC System 2.0. — М., МИФИ, Научная сессия МИФИ, 2002.
3. Rosenquist C. J. Entity Life Cycle Models and their Applicability to Information Systems Development Life Cycles // The Computer Journal. 1982. № 3. V. 25.
4. Румянцев В. П., Низаметдинов Ш. У. Проектирование сетевых моделей планирования и управления. — М., МИФИ, 1987.
5. Питерсон Дж. Теория сетей Петри и моделирование систем. — М.: Мир, 1984.
6. Jensen K. A. Brief Introduction to Coloured Petri Nets. Springer-Verlag, 1994.
7. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. — М.: ДМК, 2000.
8. Международные стандарты, поддерживающие жизненный цикл программных средств. — М.: МП "Экономика", 1996.

