

Развитие метода адаптивной кластеризации путем формирования концептуальной модели объекта информатизации

Циперман Григорий Наумович

НИТУ «МИСиС», Москва, Россия
Тел. +7(903)798-02-66
gntsip@gmail.com

Abstract. The Method of Adaptive Clustering (MAC) determines the order of formation of functional requirements and functional appearance of the projected information system. The article deals with the evolution of the MAC producing a conceptual data model from the functional model. Based on the data model, the order of building the software components of the information system is defined, which are methods encapsulated in data model objects.

Аннотация. Метод адаптивной кластеризации (МАК) определяет порядок формирования функциональных требований и функционального облика проектируемой информационной системы. В статье рассматривается развитие МАК в части продуцирования концептуальной модели данных, выводимой из функциональной модели. На основе модели данных определяется порядок построения программных компонент информационной системы, представляющих собой методы, инкапсулированные в объекты модели данных.

Ключевые слова: информационная система, метод адаптивной кластеризации, концептуальная информационная модель, сеть Петри, методы классов, МАК, UML

1 Введение

Метод адаптивной кластеризации информационных систем (МАК) – это нисходящий функциональный сервис-ориентированный метод визуального проектирования ИС, предназначенный для формирования облика создаваемой ИС как на уровне ее использования (бизнес-уровень), так и в части специфицирования ее программных компонент (кластеров). [1]

Используя метод декомпозиции бизнес-процесса, МАК определяет множество операций, которые представляют собой комплекс необходимых действий конкретного сотрудника на своем рабочем месте с целью получения полезного для бизнес-процесса промежуточного продукта.

Ключевым свойством метода является установление связей между операциями бизнес-процессов и системными сервисами (архитектурными компонента-

ми системы). Это позволяет говорить об адаптивности метода, рассматриваемой как реализация необходимой и достаточной функциональности системы, отвечающей требованиям автоматизации бизнес-процессов заказчика.

МАК определяет функциональную структуру ИС до уровня диалогов. Требуется расширить представление функциональной структуры до методов объектов, описывающих информационную структуру предметной области ИС. Для этого необходимо включить в МАК возможности определения информационной структуры ИС.

Целью работы является определение алгоритма автоматизированной идентификации объектов и связей между ними, определяющих информационную структуру предметной области бизнес-процесса с возможностью перехода к уровню концептуальной информационной модели. Объектом исследования является модель бизнес-процесса и связи функциональных объектов бизнес-процесса с объектами предметной области.

Для этого необходимо решить следующие задачи:

- Определить типы связей функциональных компонент бизнес-процесса с информационными объектами.
- Сформулировать правила вывода информационной структуры предметной области ИС из функциональной структуры бизнес-процесса.
- Описать переход от модели информационной структуры к концептуальной информационной модели предметной области.

2 Существующие методы идентификации информационных объектов

Методы идентификации объектов можно разделить на 2 группы [2]: анализ текстов и анализ задачи.

Метод анализа текстов основан на выделении существительных и глаголов в тексте описания предметной области и интерпретации их в качестве классов, атрибутов и методов. Алгоритмы, приводящие к идентификации объектов методом анализа текстов, как правило, не дают хорошие результаты. Описание метода приведено в [3], а варианты его автоматизации описаны в [4].

Метод анализа задач предполагает декомпозицию общей задачи автоматизации на иерархию задач следующих уровней. Информационные объекты возникают, как атрибуты определяемых методов (подзадач). Метод анализа задач слабо формализуем и только случайно может выявить какие-либо концептуальные информационные объекты. Методу посвящена работа [5].

3 Связи функциональных и информационных объектов

Классическое определение бизнес-процесса [6]: комплекс действий, в котором на основе одного или более видов исходных данных создается ценный для клиента результат.

Будем называть исходным ресурсом (ИР) все виды исходных данных, поступающие на вход некоторого функционального объекта (комплекса действий), результатом которого является целевой продукт (ЦП).

Будем говорить, что ЦП выводим из ИР комплексом действий и отношение между ИР и ЦП будем обозначать именованной направленной ассоциацией (Рис. 1., вариант а)¹.

Стереотип «Тип активности» в обозначении действия может принимать значения «Бизнес-функция», «Бизнес-операция», «Диалог».

Другой тип связи между ИР и ЦП соответствуют случаю, когда ЦП относится к действию, предшествующему действию, на вход которого поступает ИР (Рис. 1., вариант б). В этом случае ИР не обязательно должен совпадать с ЦП, но однозначно им определяться. В модели отношение ЦП→ИР обозначается не именованной направленной ассоциацией от ЦП к ИР с именованием ролей источника и приемника.

Будем говорить, что информационные объекты предопределены в системе (Рис. 1., вариант с), если они

- определены в моделируемом экземпляре бизнес-процесса ранее;
- возникли в рамках иных бизнес-процессов (например, справочники).

Предопределенные объекты играют роль дополнительных к основному ИР. Целевой продукт также выводим из предопределённого информационного объекта и так же в модели связь обозначается именованной направленной ассоциацией от ИР к ЦП.

4 Правила построения сети Петри, изоморфной модели бизнес-процесса

Представим функциональную модель маркированной сетью Петри с двумя типами переходов

$$M = (P, T, I, O, \mu),$$

где P – множество позиций, T – множество переходов, I, O – входная и выходная функции, отображающие переходы в комплекты позиций, μ – маркировка сети Петри.

Множество позиций сети Петри функциональной модели

$$P = (p_1, p_2, \dots, p_n), n \geq 0$$

делится на четыре непересекающихся подмножества

$$P = (N, D, K, C), \quad \cap(N, D, K, C) = \emptyset.$$

¹ Модели, представленные в данной работе, выполнены в приложении Enterprise Architect компании Sparx Systems [7].

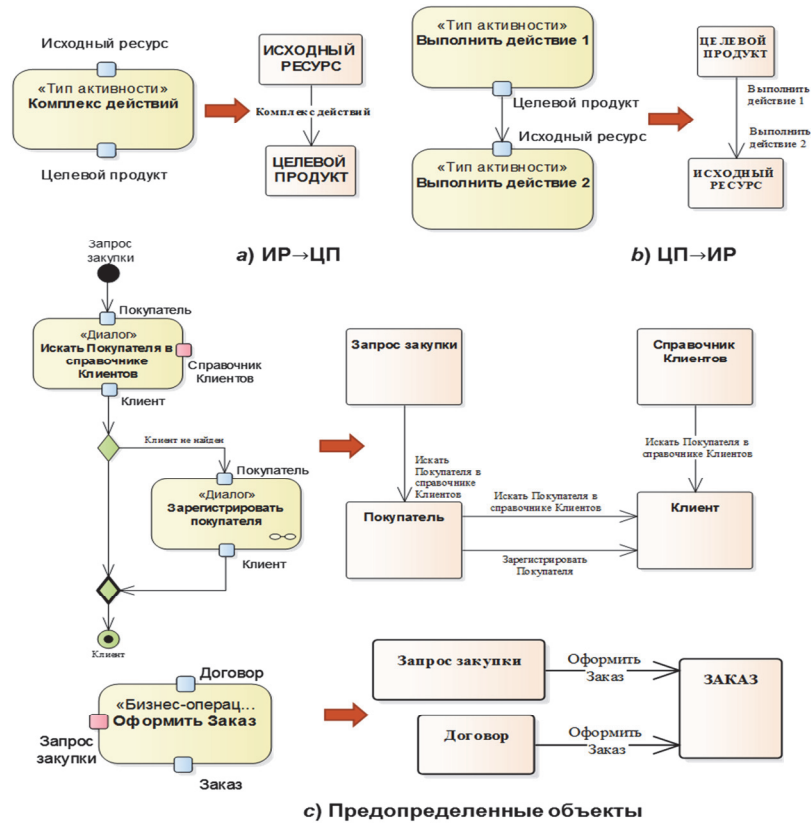


Рис. 1. Типы информационных связей ИР и ЦП

Здесь N – множество новых, еще не определенных позиций

$$N = (n_1, n_2, \dots, n_m), m \geq 0.$$

K – множество предопределенных (определенных в иных функциональных моделях), позиций:

$$K = (k_1, k_2, \dots, k_l), l \geq 0.$$

D – множество уже определенных при выполнении сети Петри позиций

$$D = (d_1, d_2, \dots, d_p), p \geq 0,$$

$$\forall t \in T: (d = I(t) \vee d = O(t)) \in D.$$

C – позиции, управляющие запуском переходов второго типа

$$C = (c_1, c_2, \dots, c_r), r \geq 0.$$

При этом, если $\tau^{(2)}$ множество переходов второго типа, то $|C| = |\tau^{(2)}|$.

Типы переходов:

$$\tau^{(1)} = (t_1^{(1)}, t_2^{(1)}, \dots, t_m^{(1)}), m \geq 0$$

$$\tau^{(2)} = (t_1^{(2)}, t_2^{(2)}, \dots, t_l^{(2)}), l \geq 0$$

Пара $\tau_i = (\tau_i^{(1)}, \tau_i^{(2)})$ определяет порядок следования переходов в модели. Тогда множество переходов сети Петри определяется как

$$T = (\tau_1, \tau_2, \dots, \tau_n), n \geq 0.$$

Входная позиция для любого типа перехода определяется как $p_i \in I(t_j^{(*)})$, а выходная – $p_i \in O(t_j^{(*)})$.

Начальная маркировка определяется как

$$\begin{aligned} \exists n_i \in N: \mu_0(n_i) = \infty; \forall n_j \in N, n_j \neq n_i: \mu_0(n_j) = 0; \\ \forall k_i \in K: \mu_0(k_i) = \infty; \\ \exists c_i \in C: \mu_0(c_i) = 1, \forall c_j \neq c_i: \mu_0(c_j) = 0. \end{aligned}$$

Начальные позиции n_i и c_i являются входными для переходов, принадлежащих одной и той же упорядоченной паре τ_i :

$$I(t_i^{(1)}) = n_i, I(t_i^{(2)}) = c_i, \text{ где } t_i^{(1)}, t_i^{(2)} \in \tau_i.$$

Перед выполнением сети Петри $D = \emptyset$. Если в ходе выполнения

$$\forall t \in T, n \in N: I(t) = n \text{ или } O(t) = n$$

то $N = N - n$, а $D = D + n$ ($n \rightarrow d$).

Условие завершения выполнения сети Петри: $N = \emptyset$ и все переходы запрещены.

Правила построения сети Петри функциональной модели для всех позиций:

1. Входные и выходные позиции сети представляют собой подмножества (не комплекты) позиций:

$$\forall p_i, p_j \in P, t \in T: \#(p_j, I(t)) = (0,1) \text{ и } \#(p_j, O(t)) = (0,1).$$

2. На вход перехода первого типа могут поступать не более одной не определенной позиции и неограниченное число других типов позиций, т.е. $\forall t \in \tau^{(1)}, n \in N, d \in D, k \in K$:

$$\#(n, I(t)) = (0,1); \#(d, I(t)) \geq 0; \#(k, I(t)) \geq 0.$$

3. На вход перехода второго типа поступает одна определенная и одна управляющая позиция, т.е. $\forall t \in \tau^{(2)}, \exists d \in D, c \in C$:

$$\#(c, I(t)) = 1; \#(d, I(t)) = 1.$$

4. Для всех типов переходов количество выходных позиций равно 1:

$$\forall t \in T: |O(t)| = 1.$$

5. В качестве выходной позиции для перехода любого типа не может быть предопределенная позиция, т.е. $\forall t \in T, k \in K$:

$$\#(k, O(t)) = 0.$$

6. Переход второго типа может сработать только один раз, так как в его входной управляющей позиции только одна фишка.

Правила именования позиций сети Петри:

- управляющие позиции не именуются
- остальные позиции получают имена соответствующих информационных объектов функциональной модели.

Переходы первого типа именуется соответствующими наименованиями действий. Переходы второго типа изображаются двумя параллельными отрезками: входным и выходным. Входной отрезок получает имя входного действия, выходной – выходного действия функциональной модели.

Рис. 2 демонстрирует правила перехода от функциональной модели к эквивалентной сети Петри.

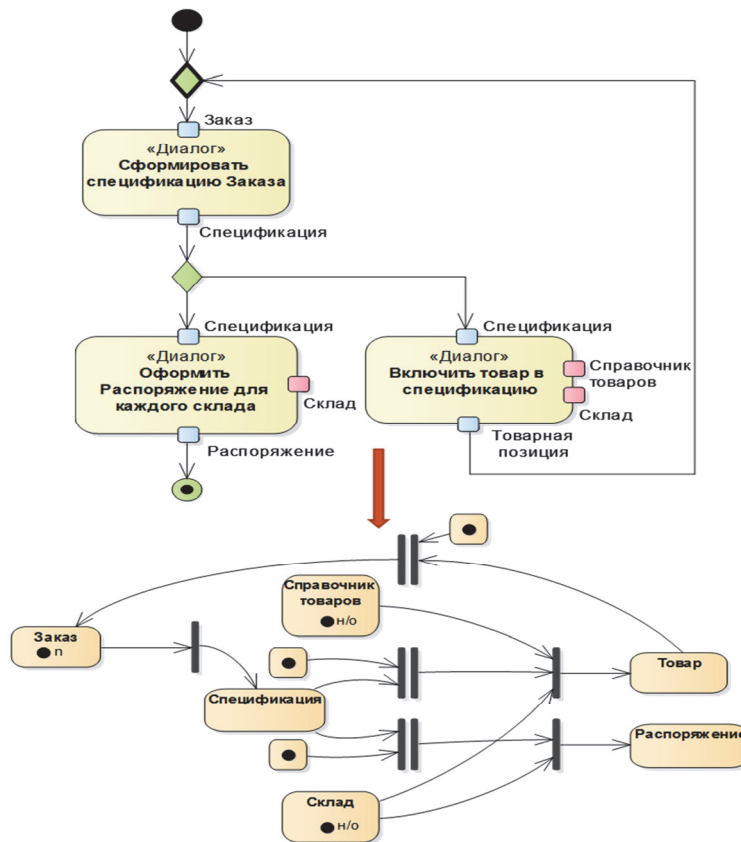


Рис. 2. Описание функциональной модели сетью Петри с двумя типами переходов

5 Формирование информационной структуры предметной области

Будем называть предварительной моделью данных модель, автоматически продуцируемую из соответствующей сети Петри. Предварительная модель данных определяет информационную структуру предметной области бизнес-процесса.

Будем исходить из того, что все программные компоненты инкапсулируются в действия, то есть программной обработки при переходе второго типа не происходит. Это означает, что формирование выходных позиций производится на уровне постусловий действия, а входных – на уровне предусловий.

С учетом этого применяются следующие правила формирования предварительной модели:

1. Переход первого типа всегда порождает связи между входной и выходной позицией.
2. Если на выходе перехода второго типа определена позиция, то связь с входной не возникает.
3. Если на выходе перехода второго типа не определена позиция, то эта позиция выводится предусловием из входной.

Алгоритм автоматического продуцирования предварительной модели данных из функциональной модели с помощью эквивалентной сети Петри выглядит следующим образом:

1. Выбрать не определенные позиции.
2. Выбрать predetermined позиции.
3. Множество определенных позиций – пустое.
4. Выбрать начальную позицию в качестве входной текущей.
5. Выбрать разрешенный переход для входной текущей позиции. Если нет разрешенных переходов – перейти на шаг 7.
 - 5.1. Выполнить переход.
 - 5.2. Выбрать не определенную позицию (входную или выходную) у выбранного перехода.
 - 5.3. Если выбранная позиция входная, то связать ее с выходной, если выходная – со всеми входными
 - 5.4. Не определенные входные и выходные позиции включить в множество определенных позиций и исключить из множества не определенных.
 - 5.5. Если переход второго типа перейти на шаг 5.6.
 - 5.5.1. Если нет входных позиций перейти на шаг 6, иначе – выбрать входную позицию
 - 5.5.2. Входную позицию связать с выходной направленной ассоциацией, присвоив ей имя перехода.
 - 5.5.3. Возврат на шаг 5.5.1.

- 5.6. Переход второго типа. Связать входную и выходную позицию направленной ассоциацией, присвоив началу имя входного отрезка перехода, а концу – выходного отрезка.
6. Перейти на шаг 4.
7. Если не пуст список не определенных позиций сообщить о некорректном построении сети Петри.
8. Конец алгоритма.

6 Формирование концептуальной информационной модели

В результате автоматического продуцирования предварительная модель данных обладает избыточным количеством связей между информационными объектами и определяет только один тип связей между ними (направленную ассоциацию).

Избыточность появляется в связи с тем, что в модели бизнес-процесса различные действия могут оперировать одними и теми же парами ИР и ЦП. При формировании предварительной модели без учета смысла этих связей невозможно определить ее обоснованность и необходимость: связь может быть ролевой, а может дублировать уже существующую. Предварительная модель данных также не может учесть характер связей между сущностями и требования к нормализации. Поэтому переход к концептуальной модели выполняется вручную.

Для преобразования предварительной модели данных в концептуальную информационную модель (**Рис. 3.**) необходимо:

- Проанализировать модель и удалить лишние связи.
- Интерпретировать и заменить типы связей на ассоциацию, генерализацию, композицию или агрегацию.
- Нормализовать концептуальную информационную модель.

Вместе с тем, применение алгоритма продуцирования предварительной модели данных позволило поднять качество и сократить временные затраты в процессе анализа сложных информационных систем за счет:

- сфокусированного выделения информационных объектов в процессе моделирования автоматизируемого бизнес-процесса,
- минимизации затрат на процесс идентификации объектов предметной области и связей между ними,
- и, как следствие, сокращения проектных итераций.

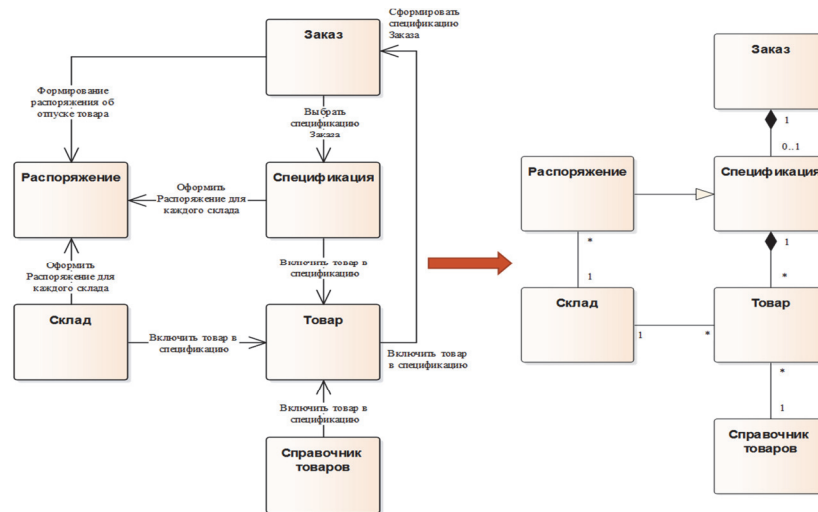


Рис. 3. Преобразование предварительной модели данных в концептуальную информационную модель

Литература

1. Г. Н. Циперман, «Применение метода адаптивной кластеризации для проектирования сложных информационных систем,» в *III Научно-практическая конференция «Актуальные проблемы системной и программной инженерии». Сборник научных трудов*, Москва, 2013.
2. И. Грэхем, *Объектно-ориентированные методы. Принципы и практика*, Москва: Издательский дом "Вильямс", 2004.
3. Abbot R.J., «Program Design by Informal English Descriptions» *Communications of the ACM*, т. 26, № 11, pp. 882-894, 1983.
4. Saeki M., Horai H. и Enomodo H., «Process from Natural Language Specification» в *Proceedings of the 11th International Conference on Software Engineering Software Development*, New-York, 1989.
5. Daniels P.J., «The user modelling function of an intelligent interface for document retrieval systems» *Intelligent Information Systems for the Information Society*, pp. 162-176, 1986.
6. М. Хаммер и Д. Чампи, *Реинжиниринг корпорации. Манифест революции в бизнесе*, Москва: Манн, Иванов и Фербер, 2006.
7. Sparx System Pty Ltd, «Enterprise Architect» [В Интернете]. Available: <http://sparxsystems.com/products/ea>.