

УФО-АНАЛИЗ И МОДЕЛИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ЗАПАСАМИ

МАТОРИН С.И.

Рассматривается пример использования метода объектно-ориентированного системологического анализа (УФО-анализа) для моделирования и проектирования информационной системы поддержки однопродуктового складского хозяйства.

Алгоритм УФО-анализа, представленный в работе [1], является развитием и формализацией метода системологического анализа *OM SAD* [2], согласующегося с процедурами объектно-ориентированного анализа и проектирования (ООА и ООД). Алгоритм основан на концептуальной классификационной модели “Узлы – Функции – Объекты” [1] и позволяет использовать формализованные правила выявления классов и объектов предметной области в процессе ООА и ООД. Работы по развитию, применению данного метода и построению на его основе инструментальных CASE-средств проводятся в рамках нового научного направления системологического анализа сложных динамических объектов [3].

Кратко суть данного алгоритма может быть представлена следующими основными шагами:

- выявление узлов связей в структуре разрабатываемой системы на основании ее функциональных связей в целом, задаваемых заказчиком;
- выявление функциональности, поддерживающей (обеспечивающей) обнаруженные узлы;
- определение объектов, соответствующих выявленной функциональности.

При этом первый шаг может быть отождествлен с этапом анализа системы, второй – с этапом ее проектирования, а третий – с ее реализацией.

Одной из привлекательных особенностей алгоритма является возможность автоматизации его шагов путем использования *формально-семантического алфавита*, на котором основана нормативная система, описывающая данный метод системного анализа [1,2]. При этом иерархия классов “Узлы – Функции – Объекты” должна быть адаптирована к решаемой задаче. Для этого классы упомянутой иерархии должны быть предварительно специализированы в конкретной предметной области.

Рассмотрим практический пример проведения УФО-анализа и моделирования.

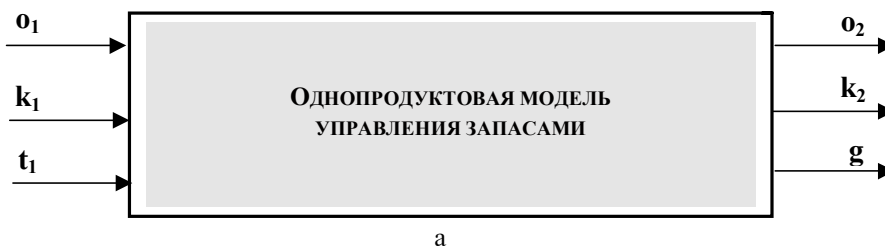
Для этого возьмем простую информационную систему поддержки однопродуктового складского хозяйства, аналогичную однопродуктовой системе управления запасами, приведенной в работе [4].

На складе хранится некий товар, который отправляется (**O2**) в магазины по их заказам (**K1**). Склад для пополнения своих запасов (в случае их снижения ниже определенного уровня – **H**) может делать заявки (**K2**), по которым он получает товар (**O1**) из центрального источника. Известны стоимость доставки единицы товара (\$2), себестоимость хранения единицы товара за каждый день (\$3), плата за неотгрузку единицы товара по заказу (\$4). Требуется построить информационную систему, позволяющую управлять запасами данного склада, путем определения и минимизации общих складских издержек (**g**) в зависимости от уровня неснижаемых запасов и объемов поставок из центра.

Результаты первичной модификации (адаптации) базовой классификации связей [1], построения контекстной объектной модели и составления предварительной таблицы узлов разрабатываемой системы в соответствии с алгоритмом [1] представлены на рис.1. При этом строки таблицы (б) узлов соответствуют входящим связям, столбцы – выходящим. На рис. 1 показаны только внешние функциональные связи системы, отражающие постановку задачи на разработку. Комбинирование этих связей, в данном случае, не позволяет выявить узлы в структуре системы, которые можно было бы обеспечить известной функциональностью. Это вынуждает перейти, в соответствии с алгоритмом, к итерационному процессу добавления (в классификацию и таблицу) новых (поддерживающих) связей и поиска узлов с реализуемой функциональностью.

Результаты промежуточного этапа добавления поддерживающих связей, обнаружения узлов на первом внутреннем уровне иерархии системы и построения ее промежуточной объектной модели с соответствующей функциональностью представлены на рис. 2. Процесс добавления новых связей не лишен полностью творческого начала, однако направляется он не только здравым смыслом, но и известными конкретными исходными данными (связями, которые надо замкнуть), а также необходимостью получения в узлах понятной (реализуемой) функциональности. При этом необходимо иметь в виду, что все поддерживающие (внутренние) связи, очевидно, будут и входящими (в строках), и выходящими (в столбцах).

Система поддержки складского хозяйства, очевидно, должна отслеживать поток получаемых, хранящихся и отправляемых товаров. Это позволяет добавить очевидную поддерживающую связь **Ox** (хранящиеся товары), являющуюся результирующей потока **O1** поступающих товаров. Именно из потока **Ox** с учетом объема заказов **K1** на отправку в магазины формируется поток **O2**, т.е. происходит отправка товаров. Следующими очевидными внутренними (поддерживающими) связями являются связи **g2**, **g3** и **g4**, которые представляют собой суммы соответствующих складских расходов (за доставку, за хранение и за неотгрузку).

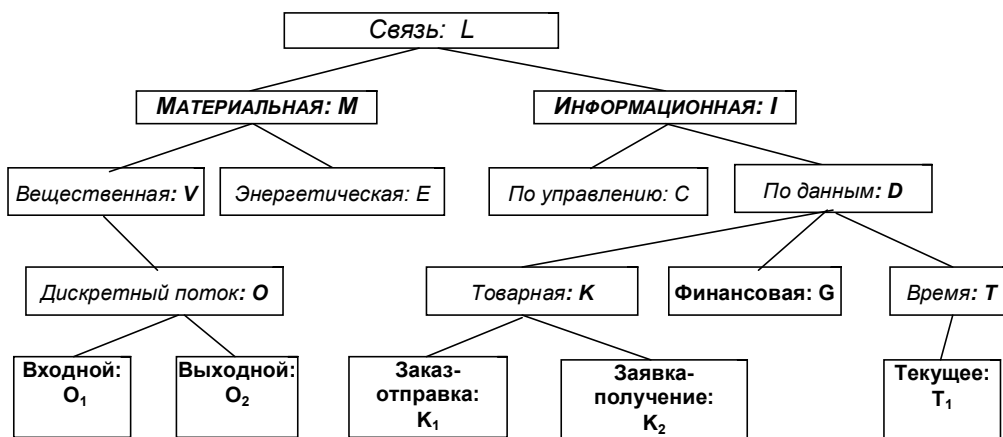


а

УЗЛЫ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ

			o_2						
				k_2					
					g				
o_1									Получение товара
	k_1								Заказ на отправку в магазин
		t_1							Время (смена даты)
			Отправка товара	Заявка на пополнение запасов	Складские издержки				

б



в

Рис. 1. Функциональные связи системы: а – контекстная модель системы управления запасами; б – таблица “Узлы”, составленная из функциональных связей системы; в – модифицированная классификация связей

Естественно, что $g = g_2 + g_3 + g_4$. Однако только относительно потока g_2 можно точно сказать, что он является результирующим потоком K_2 (т.е. поток K_2 должен быть не только внешним, но и внутренним) и самого себя (для обеспечения подсчета платы за доставку “нарастающим итогом” ввиду того, что за время моделирования доставка могла осуществляться неоднократно), так как по условию склад тогда и столько получает, когда и сколько заказывает. Последней очевидной поддерживающей связью на данном уровне будет, видимо, связь K_1 (регистрация поступления единицы товара), являющаяся производной от потока хранящихся товаров O_1 . Данная информация необходима ввиду того, что склад платит за хранение каждой единицы в сутки. Кроме того, для расчета платы за неотгрузку необходима информация о количестве хранящихся товаров, которое зависит, естественно, и от числа поступивших товаров.

Функции, соответствующие выявленным на данном этапе узлам, представлены в таблице (№1 – №5). При этом остается неясным откуда и каким образом появляется поток K_2 , т.е. когда и какие подаются заявки на пополнение запасов склада. Кроме того, до конца не ясен весь процесс определения издержек склада, так как не понятно, каким образом появляются потоки g_3 и g_4 . Задача, однако, существенно сузилась и конкретизировалась, так как выявлен с конкретными входами и выходами внутренний блок (подсистема), который отвечает за формирование пока не завязанных в узлы потоков.

Результаты завершающего этапа анализа и проектирования, на котором выявляются все узлы системы и функции, а также строится объектная модель разрабатываемой системы, представлены на рис. 3.

При этом связь K_X , очевидно, есть результирующая потоков K_{01} (добавление единицы при поступлении единицы товара) и K_1 (количество отгружаемого товара).

Функции, соответствующие выявленным на данном этапе узлам, представлены в таблице (№6 – №9).

Таким образом, все структурные (узловые) и функциональные характеристики системы оказываются выявленными. Процесс проектирования завершен. Далее следует, в соответствии с алгоритмом [1], имитационное моделирование (требующее, однако, использования специального инструментария) и поиск подходящих объектов, реализующих полученную в узлах функциональность.

Функции элементов

№	ФОРМАЛЬНО:	СОДЕРЖАТЕЛЬНО:
1	$o_X = F(o_1)$	Существуют стандарты на способы хранения всех видов товаров. Кроме того, данная функция непосредственно разрабатываемой информационной системой не отслеживается. Однако она необходима в модели для обеспечения связи с объектом управления
2	$o_2 = F(o_X, k_1)$	Также существуют стандартные способы отгрузки всех видов товаров. В модели эта функция учтена в связи с тем, что разрабатываемая система должна контролировать связь k_1
3	$k_{01} = F(o_X)$	Регистрация товара путем считывания специальным сканером кода продукта. Оборудование и программное обеспечение такого регистрирующего терминала существует в различных модификациях
4	$g_2 = g_2 + S_2 * k_2$	Подсчет «нарастающим итогом» издержек в виде оплаты за доставленные товары по заявке
5	$g = g_2 + g_3 + g_4$	Без комментариев
6	$k_X = F(k_{01}, k_1)$ Например: $k_X = k_X + 1 - k_1$ (при каждой регистрации k_{01})	Данный узел может быть обеспечен функциональностью различными способами. Например, при регистрации может осуществляться просто занесение данных о поступившей единице товара в БД (таблицу, очередь). Тогда число строк в таблице (элементов в очереди) и будет k_X , с учетом удаления k_1 отгружаемых товаров
7	$k_X - H \leq 0 \rightarrow k_2$	Без комментариев
8	$k_1 - k_X \geq 0 \rightarrow g_4 = g_4 + (k_1 - k_X) * S_4$	Плата за неотгрузку «нарастающим итогом» вычисляется, если заказ магазина превышает число хранящихся товаров
9	$g_3 = g_3 + k_X * S_3$	Определение платы за текущий день за все хранимые товары и «нарастающим итогом» себестоимости хранения товаров за время моделирования

Главной особенностью представленного алгоритма является возможность автоматизации обнаружения узлов в структуре системы, поиска функций, обеспечивающих выявленные узлы, и идентификации объектов, реализующих требуемую функциональность, путем использования иерархии классов «Узлы – Функции – Объекты».

Например, в данном случае, при разработке информационной системы поддержки складского хозяйства до начала разработки или в процессе анализа требований заказчика исполнители могли (и должны были) включить в классификацию узлов узлы типа $(O_1)O_X$ (принятие товаров на хранение), $(O_X)K_{01}$ (регистрация единиц товара), $(K_{01}, K_1)K_X$ (организация элементов в очередь и ее модификация с вычислением длины очереди), $(g_2, g_3, g_4)g$ (суммирование расходов) и т.д. В классификации функций для каждого из этих узлов должна существовать функция (в общем случае не единственная) со всеми формальными и содержательными характеристиками. В классификации элементов для каждой функции должен существовать специфицированный объект (в общем случае не один). Тогда, естественно, при условии использования тех символов, которыми обозначены конкретные классы в иерархии «Узлы – Функции – Объекты», программная система поддержки данного метода анализа будет предлагать наборы, соответственно, функций для опознаваемых ею узлов и объектов для опознаваемых ею функций.

Выводы

Предложенный алгоритм УФО-анализа, представляющий собой развитие и формализацию метода системологического анализа OMSAD, позволяет использовать формализованные правила выявления классов и объектов предметной области в процессе объектно-ориентированного анализа и проектирования.

Полученные результаты положены в основу разработок CASE-средств нового поколения, облегчающих анализ и проектирование сложных систем и предназначенных для использования как аналитиками-профессионалами, так и специалистами различных предметных областей.

Литература: 1. Маторин С.И. УФО-анализ // АСУ и приборы автоматики. 2001. Вып 117. С.5-13. 2. Маторин С.И. О новом методе системологического анализа, согласованном с процедурой объектно-ориентированного проектирования. Ч.1 // Кибернетика и системный анализ. 2001. №4. С.119-132. 3. Маторин С.И. О новом научном направлении системологического анализа сложных динамических объектов // Вестник ХГПУ. Новые решения в современных технологиях. 2000. №81. С.15-18. 4. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. М.: «АНВИК», 1998. 427 с.

Поступила в редколлегию 14.09.2001

Рецензент: д-р техн. наук, проф. Авраменко В.П.

Маторин Сергей Игоревич, канд. техн. наук, ст. науч. сотр., докторант ХНУРЭ. Научные интересы: системные знаниеориентированные технологии. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-95-91, 47-41-85.