© Д.В. Александров, А.В. Костров, 2008

УДК 658.512

Д.В. Александров, А.В. Костров

ПРОЦЕДУРА ПЛАНИРОВАНИЯ ПРОЦЕССА ИЗГОТОВЛЕНИЯ СЛОЖНОГО ПРОИЗВОДСТВЕННОГО ЗАКАЗА

При планировании выполнения сложного и объемного заказа в условиях *многономенклатурного мелкосерийного и единичного производства* (ММЕП), прежде всего, формируется маршрутно-сборочная схема, обычно оформляемая в виде «дерева» заказа. Значение сборочной схемы для *оперативно-производст-венного планирования* (ОПП) заключается в том, что она наглядно показывает порядок и последовательность включения первичных элементов, или *номенклатурных позиций* (НП), сложного составного информационного продукта в сборочные агрегаты, или узлы, отражая отношения принадлежности и входимости в структуре заказа. Тем самым схема определяет возможность параллельного выполнения разных узлов заказа. Распараллеливанием работ, в частности, достигается значительное сокращение *длительности производственных циклов* (ДПЦ). Представленная в таком виде схема является структурной характеристикой производственного заказа и является основой при формировании календарного циклового графика изготовления заказа. Процесс изготовления заказа представляется на цикловом графике в виде множества технологических цепочек простых производственных процессов, начальные звенья которых соответствуют изготовлению НП, находящихся на нижнем уровне входимости в структуре заказа, а конечные – финишному процессу, в котором создается готовый продукт. Каждой отдельной НП в составе заказа соответствует календарный отрезок времени на цикловом графике, в течение которого она должна быть изготовлена или закуплена, при этом начало изготовления – так называемое *опережение* – регламентируется циклом ее изготовления, а также сроком опережения сборочной единицы, в которую она входит.

Как известно, изготовление и выпуск продукции в ММЕП осуществляется на заказ. Множество, предс-тавляющее производственный заказ, можно задать следующим образом [1]:

$$Z = \left\{ Z^A, P \right\},$$

где Z^A – множество атрибутов производственного заказа; P – множество НП производственного заказа. Под НП производственного заказа понимается такой его компонент, для которого определены количество, необходимое для комплектации родительской сборочной единицы, уровень входимости, а также порядковый номер, определяющий местоположение компонента в спецификации заказа. Множество НП производственного заказа, их количество, а также порядок, в соответствии с которым они располагаются в заказе, называется спецификацией заказа, или составом заказа.

Множество НП производственного заказа имеет вид:

$$P = \left\{ P_i \mid i = \overline{1, I^P} \right\},$$

где I^P – количество НП в производственном заказе; P_i – множество, представляющее *i*-ю НП производственного заказа, которое имеет вид:

$$P_i = \left\{ P_i^A, \tilde{P}^i \right\},$$

где $\tilde{P}^i$ – множество, представляющее состав *i*-й НП; P_i^A – множество атрибутов *i*-й НП, которое имеет вид:

$$P_i^A = \left(P_{i1}^A, P_{i2}^A, P_{i3}^A, P_{i4}^A, P_{i5}^A, P_{i6}^A, P_{i7}^A, P_{i8}^A, P_{i9}^A, \ldots \right),$$

где P_{i1}^A – порядковый номер НП в спецификации заказа; P_{i2}^A – уровень входимости НП в заказе; P_{i3}^A – код компонента; P_{i4}^A – наименование компонента; P_{i5}^A – требуемое количество единиц компонента данной НП; P_{i6}^A – цикл изготовления или закупки НП; P_{i7}^A – опережение запуска НП; P_{i8}^A – изготавливаемое количество единиц компонента; P_{i9}^A – порядковый номер *i*-й НП, в которой будет изготавливаться компонент.

Множество, представляющее состав *i*-й позиции, имеет вид:

$$\tilde{P}^i = \left\{ \tilde{P}_j^i \mid j = 1, J^{\tilde{P}^i} \right\},$$

где $J^{\tilde{P}^i}$ – количество дочерних НП в составе *i*-й НП; $\tilde{P}_j^i$ – множество, представляющее входимость *j*-й дочерней НП в *i*-ю НП, имеющее вид

$$\tilde{P}_j^i = \left(\tilde{P}_{j1}^i, \tilde{P}_{j2}^i, \ldots \right),$$

где $\tilde{P}_{j1}^i$ – код *j*-й НП; $\tilde{P}_{j2}^i$ – количество, в котором *j*-я НП входит в *i*-ю НП.

Множество НП независимого спроса можно представить в виде

$$P^0 = \left\{ P_y^0 \mid y = 1, Y^{P^0} \right\},$$

где P_y^0 – *y*-я НП верхнего уровня входимости; Y^{P^0} – количество НП верхнего уровня входимости, причем $P^0 = P \setminus \bigcup \tilde{P}^i$, где $i = \overline{1, I^P}$.

Для описания состава заказа вводится множество $R = \left\{ R_i \mid i = \overline{1, I^R} \right\}$, где I^R – количество элементов в этом множестве, R_i – множество, представляющее *i*-й элемент во множестве R. Причем количество элементов во множестве R равно количеству НП верхнего уровня входимости $I^R = Y^{P^0}$, а каждый i-й элемент множества R является НП с индексом *y* множества P^0 : $R_i = P_y^0$, где $i = \left| P^0 \right| - y + 1$.

Для разработки алгоритма структурирования заказа и планирования опережений для НП нужно ввести следующие вспомогательные функции:

- $P_y^0 \xrightarrow{f_{P_y^0}} P_i$ – определяет НП P_i, которой соответствует НП P_y^0 во множестве НП верхнего уровня входимости;
- $R_i \xrightarrow{f_{R_i}} P_j$ – определяет НП спецификации заказа P_j, которая соответствует *i*-й НП множества R;
- $\tilde{P}_j^i \xrightarrow{f_{\tilde{P}_j^i}} P_h$ определяет НП спецификации заказа P_h, которая соответствует *j*-му элементу множества $\tilde{P}^i$;
- $P_i \xrightarrow{f_{P_i^{rod}}} P_j$ – определяет НП спецификации заказа P_j, которая является родительской по отношению к НП P_i.

В случае, если НП P_i является НП верхнего уровня входимости, то

есть $\exists P_l^0 \in P^0 : f_{P_i^0} = P_i$, функция $f_{P_i^{rod}}$ возвращает значение 0: $f_{P_i^{rod}} = 0$.

Введем переменные: *z* – переменная, определяющая порядковый номер НП в спецификации заказа; *u* – переменная, определяющая уровень входимости НП в спецификации заказа; *v* – требуемое количество единиц компонента в спецификации заказа; *w* – счетчик дочерних НП в составе соответствующей родительской НП. Далее с использованием введенных функций предлагается алгоритм планирования опережений и состава заказа.

1. Установить: $z = 0$.

2. Каждой НП спецификации верхнего уровня входимости $\forall P_y^0 \in P^0, \exists P_v \in P : P_v = f_{P_y^0}$ присвоить нулевой уровень входимости: $P_{v2}^A = 0$.

3. Выполнить действия, перебирая НП R_i из множества R, начиная с последнего элемента: $R_i, i = |R|$.

3.1. Определить НП спецификации заказа P_c, соответствующую эле-менту $R_i : \exists P_c \in P : P_c = f_{R_i}$.

3.2. Присвоить значение переменной v, равное требуемому количеству единиц компонента НП P_c : $v = P_{c5}^A$; присвоить значение переменной u, равное уровню входимости НП $P_c : u = P_{c2}^A$.

3.3. Если НП P_c не имеет состава, перейти к шагу 3.4.: $\tilde{P}^c = \{\ \}, P_c = \{P_c^A, \tilde{P}^c\}$.

3.3.1. Перебирать состав НП P_c. Перебор осуществляется по убыванию значения индекса элемента множества $\tilde{P}^c : \tilde{P}_w^c, w = \overline{|\tilde{P}^c|, 1}$.

3.3.1.1. НП спецификации заказа P_n : $\exists P_n, P_m \in P : P_n = f_{P_m^{rod}}, P_m = f_{\tilde{P}_w^c}$ добавить во множество $R, R = R \cup \{P_n\}$.

3.3.1.2. Рассчитать значение требуемого количества единиц НП $P_n : P_{n5}^A = v \cdot \tilde{P}_{w2}$; установить значение уровня входимости НП P_n, равное $u+1 : P_{n1}^A = u+1$.

3.3.1.3. Уменьшить значение счетчика НП состава на 1: $w = w - 1$.

3.3.2. Перейти к шагу 3.3.1 если $w \neq 0$.

3.4. Увеличить значение переменной z на единицу: $z = z + 1$.

3.5. Присвоить значение порядкового номера НП P_c в спецификации заказа, равное значению переменной z : $P_{c1}^A = z$; установить порядковый номер НП в спецификации заказа, в котором будет изготавливаться НП P_c, равный номеру этой НП: $P_{c9}^A = P_{c1}^A$; установить значение изготавливаемого количества единиц компонента НП P_c, равное требуемому количеству: $P_{c8}^A = P_{c5}^A$.

3.6. Выполнить процедуру расчета ДПЦ НП P_c.

3.7. Выполнить процедуру расчета опережения запуска НП P_c.

3.8. Удалить элемент R_i из множества $R : R = R \setminus \{R_i\}$.

4. Перейти к шагу 3, если $R \neq \{\ \}$, иначе завершить алгоритм.

Результатом работы алгоритма является структурированный состав заказа, для каждой НП которого установлен срок опережения запуска в производство. При выполнении функции календарного планирования ука-

зывается либо дата запуска заказа в производство, либо дата его изготовления. При необходимости пользователем могут изменяться норматив межоперационного пролеживания НП, коэффициент выполнения норм и др. Результатом календарного планирования является график изготовления заказа в виде циклограммы. Следует отметить, что работа представленного алгоритма в целом соответствует логике планирования в системах класса *MRP* II с учетом специфики ММЕП. Кроме того, данный алгоритм позволяет выполнять корректировки плана с учетом объединения НП в партии.

СПИСОК ЛИТЕРАТУРЫ

1. *Костров, А.В.* Методы и модели информационного менеджмента: учеб. пособие / Д. В. Александров, А. В. Костров, Р.И.Макаров, Е.Р. Хорошева; под ред. А.В. Кострова. – М.: Финансы и статистика, 2007. – 336 с.: ил. – ISBN 978-5-279-03067-5. ГИАБ

Коротко об авторах

Александров Д.В. – докторант, кандидат технических наук,
Костров А.В. –
Владимирский государственный университет, г. Владимир.

Статья представлена Владимирским государственным университетом.

© Д.В. Александров, 2008

УДК 658.512

Д.В. Александров

ФОРМАЛИЗАЦИЯ И РАЗВИТИЕ МЕТОДОЛОГИИ IDEF0 ДЛЯ СИСТЕМНОГО МОДЕЛИРОВАНИЯ ОРГАНИЗАЦИЙ

При построении функциональной модели бизнес-про-цессов промышленного предприятия (в общем случае – организации) необходимо выполнить значительный объем работ по формализации описания этого объекта. На основе имеющейся об объекте информации необходимо создать его *IDEF*0-модель как модель данной предметной области с требуемой степенью детализации и поддерживать ее в актуальном состоянии на протяжении всего времени существования предприятия. Полная модель подобной сложной системы может включать в себя тысячи

функциональных блоков и отношений, поэтому требует для создания значительных трудовых затрат. При этом обычно привлекается группа системных аналитиков, каждый из которых прорабатывает определенную ветвь глобальной иерархической модели – субмодель [1, 2].

Глобальная *IDEF*0-модель содержит в себе конечные множества всех функциональных блоков и отношений, а все остальные *IDEF*0-модели являются субмоделями глобальной. Кроме того, каждая субмодель имеет контекстную диаграмму, точку зрения и цель, что придает ей статус самостоятельной модели. Из этого следует, что существует множество контекстных диаграмм, содержащих цели и точки зрения. Таким образом, работая с функциональной моделью определенной предметной области, например, какого-либо биз-нес-процесса предприятия, аналитик лишь декомпозирует соответствующий функциональный блок глобальной модели, отождествляя его с соответствующей контекстной диаграммой.

Как отмечалось выше, необходимо не только создать модель, но и поддерживать ее в актуальном состоянии. Известно, что одной из основ деятельности предприятия является его удачная, соответствующая времени, целям и задачам организационная структура (ОС). Внешняя среда организации очень часто меняется, поэтому необходимо постоянно корректировать ОС и соответствующую ей модель, в то время как звенья структуры остаются постоянными. Для решения этих задач могут эффективно использоваться *IDEF*0-модели [1, 2]. Ниже приведено формализованное описание методологии.

Системы IDEF0-моделей. В сложных системах целесообразно обобщить множество моделей по отношению к предметной области в виде системы моделей. Система *IDEF*0-моделей – это множество моделей, описывающих некоторую предметную область с различных точек зрения на основе общих базовых наборов объектов [1, 2]:

а) преобразуемых системой (входов);

б) управляющих выполнением этих преобразований (управлений);

в) изображающих физические аспекты функций (механизмов).

Каждая система *IDEF*0-моделей (в дальнейшем просто система) является независимым проектом, представляющим собой совокупность моделей, описывающих деятельность какой-либо организации. Здесь и далее под моделью понимается *IDEF*0-модель. Таким образом, каждая модель описывает один конкретный бизнес-процесс, а система моделей служит для представления всей совокупности бизнес-процессов предприятия, их различных вариантов и версий. Множество систем моделей Z имеет вид:

$$Z = \left\{ Z_x \middle| x = \overline{1, X} \right\},$$

где Z_x – множество, представляющее *x*-ю систему моделей; *X* – количество систем моделей. Здесь

$$Z_x = \left\{ Z_x^A, \dot{M}_x \right\},$$

где Z_x^A – множество атрибутов *x*-й системы; $\dot{M}_x$ – множество моделей *x*-й системы. Множество атрибутов *x*-й системы имеет вид:

$$Z_x^A = \left(Z_{x1}^A, \ldots \right),$$

где Z_{x1}^A – уникальное наименование системы моделей среди существующих систем: $\forall Z_x \in Z, Z_x \notin \chi$.

*Модели. IDEF*0-модель – это иерархически организованная совокупность диаграмм, состоящих из блоков, соеди-

ненных дугами, где верхняя диаграмма является наиболее общей, а нижние диаграммы наиболее детализированы. Множество моделей некоторой *x*-й системы имеет вид:

$$\dot{M}_x = \left\{ \dot{M}_{xy} \middle| y = \overline{0, Y_x} \right\},$$

где $\dot{M}_{xy}$ – множество, представляющее в *x*-й системе *y*-ю модель; Y_x – количество моделей в *x*-й системе.

Модель с нулевым индексом $\dot{M}_{x0}$ называется системной моделью, она создается вместе с системой, имеет то же название, что и система, и может быть удалена только вместе с ней. Далее необходимо определить элементы, из которых состоит любая модель, поэтому для упрощения описания представляющих их множеств потребуется ввести обозначение для модели любой системы:

$\dot{M}_{xy} = M$, тем самым избавляясь от индексов *x* и *y*.

Далее и для других множеств будут использоваться аналогичные обозначения, поэтому необходимо ввести несколько общих положений, регламентирующих использование вариантов обозначения множеств:

а) какое-либо множество, обозначенное латинской буквой с точкой без индексов, обозначает, что оно является базовым, т.е. его элементы принадлежат различным системам моделей;

б) какое-либо множество, обозначенное латинской буквой с точкой и индексом *x*, говорит о том, что каждый элемент этого множества относится к *x*-й системе;

в) какое-либо множество, обозначенное латинской буквой с точкой и индексами *x* и *y*, означает, что все его элементы относятся к *y*-й модели *x*-й системы;

г) какое-либо множество, обозначенное латинской буквой со штрихом и индексом *x*, говорит о том, что каждый элемент этого множества имеет сквозной уникальный индекс в *x*-й системе, несмотря на то что этот элемент является элементом какой-либо модели в этой системе.

Таким образом, множество, представляющее модель M, имеет вид:

$$M = \left\{ M^A, S \right\},$$

где M^A – множество атрибутов модели; S – множество субмоделей модели.

Множество атрибутов модели должно быть представлено в виде:

$$M^A = \left(M_1^A, \ldots \right),$$

где M_1^A – уникальное наименование модели: $\forall \dot{M}_{xy} \in \dot{M}_x, \dot{M}_{xy} \notin \chi$.

У каждой модели системы имеется владелец представляемого этой моделью бизнес-процесса. Каждая *y*-я модель *x*-й системы представлена в виде блока на системной диаграмме $\dot{D}_{x0}$ (контекстного блока модели).

Субмодели. Субмодель – это определенная ветвь модели, имеющая собственную точку зрения (которая принимается на основании точки зрения модели и не противоречит ей), цель, являющуюся одной из подцелей дерева целей модели и контекстную диаграмму, содержащую блок, расположенный на одной из диаграмм модели. Множество субмоделей S необходимо описать так:

$$S = \left\{ S_i \middle| i = \overline{0, I^S} \right\},$$

где S_i – множество, представляющее *i*-ю субмодель при $i = \overline{1, I^S}$ или собственно модель при $i = 0$ (нулевую или контекстную субмодель); I^S – количество субмоделей в модели. Контекстная суб-

модель создается вместе с моделью, и в случае, когда модель состоит из единственной контекстной субмодели, множество S содержит единственный элемент: $S = \{S_0\}$. Системная модель состоит из единственной контекстной (системной) субмодели.

Учитывая тот факт, что модель рассматривается в виде совокупности субмоделей, можно дать и другое определение модели:

IDEF0-модель – это иерархически организованная совокупность субмоделей, каждая из которых обладает следующими признаками:

а) состоит из диаграмм;

б) имеет собственную точку зрения, не противоречащую модели;

в) базируется на цели, являющейся подцелью цели модели;

г) представляется собственной контекстной диаграммой с блоком, отождествленным с одним из блоков модели, причем верхняя контекстная субмодель является обобщенной, а нижние субмодели воспроизводят лишь узкие проблемные области.

В общем случае субмодель служит для описания субпроцессов конкретного бизнес-процесса предприятия. Множество, представляющее *i*-ю субмодель, должно иметь вид:

$$S_i = \left\{S_i^A, D_i^S, \tilde{S}_i\right\}, \qquad (*)$$

где S_i^A – множество атрибутов *i*-й субмодели; D_i^S – множество диаграмм, из которых состоит *i*-я субмодель, исключая диаграммы, составляющие субмодели-потомки *i*-й субмодели; $\tilde{S}_i \subset S$ – множество субмоделей-потомков *i*-й субмодели, причем:

$$S \setminus \bigcup_{i=0}^{I^S} \tilde{S}_i = S_0,$$

$$\forall a \in \tilde{S}_q, \tilde{S}_q \subset S, q = \overline{0, I^S},$$

$$\forall b \in \tilde{S}_v, \tilde{S}_v \subset S, v = \overline{0, I^S}:$$

$$q \neq v \Rightarrow a \neq b.$$

Каждая субмодель имеет собственную контекстную диаграмму $D_{i0}^S \in D_i^S$. А для системной субмодели эта контекстная диаграмма является единственной, она также называется контекстной диаграммой системы.

Множество атрибутов *i*-й субмодели S_i^A должно быть представлено следующим образом:

$$S_i^A = \left(S_{i1}^A, S_{i2}^A, S_{i3}^A, \ldots\right),$$

где S_{i1}^A – уникальное наименование субмодели: $\forall S_i \in S, S_i \notin \chi$; S_{i2}^A – точка зрения субмодели; S_{i3}^A – цель субмодели, которая также включает индикатор ее достижения. Необходимо также ввести обозначение.

Диаграммы. Диаграмма – это согласованное описание фрагмента предметной области, состоящее из блоков, соединенных дугами. Множество диаграмм модели D имеет вид:

$$D = \left\{D_i \middle| i = \overline{0, I^D}\right\},$$

где D_i – множество, представляющее *i*-ю диаграмму модели; I^D – количество диаграмм в модели.

$$D_i = \left\{D_i^A, V_i^D, L_i^D, W_i^D\right\},$$

где D_i^A – множество атрибутов *i*-й диаграммы; V_i^D – множество вершин, расположенных на *i*-й диаграмме; L_i^D – множество линий, соединяющих вершины *i*-й диаграммы; W_i^D – множество комментариев *i*-й диаграммы.

Множество атрибутов *i*-й диаграммы D_i^A должно быть задано так:

$$D_i^A = \left(D_{i1}^A, D_{i2}^A, \ldots\right),$$

где D^A_{i1} – уникальное наименование диаграммы; D^A_{i2} – уникальный *C*-номер диаграммы. Фактически из (*) следует, что множество D разбивается на подмножества в соответствии с принадлежностью диаграмм субмоделям:

$$\bigcup_{i=0}^{I^S} D^S_i = D .$$

Кроме того, следует разбить множество диаграмм на два подмножества:

а) $D^C \subseteq D$ – множество контекстных диаграмм;

б) $D^R \subseteq D$ – множество регулярных диаграмм.

Как отмечалось выше, каждая система моделей имеет единственную контекстную диаграмму системы. Контекстная диаграмма системы моделей (системная диаграмма) – это диаграмма, на которой представлены модели этой системы (модельные блоки), вся окружающая среда в виде единственного внешнего блока, а дуги, соединяющие модельные блоки между собой и с внешним блоком, представляют соответственно взаимосвязи моделей и отношения их с внешней средой [1, 2]. С учетом принятой выше системы обозначений множество диаграмм *x*-й системы моделей имеет вид:

$$\bigcup_{y=0}^{Y_x} \dot{D}_{xy} = \dot{D}_x ,$$

где $\dot{D}_{xy}$ – множество диаграмм *y*-й модели.

$$\dot{D}_x = \left\{ D'_{xi} \middle| i = 0, I^D_x \right\} ,$$

где D'_{xi} – множество, представляющее *i*-ю диаграмму в *x*-й системе, I^D_x – количество диаграмм в *x*-й системе:

$$I^D_x = \sum_{y=0}^{Y_x} I^D_{xy} ,$$

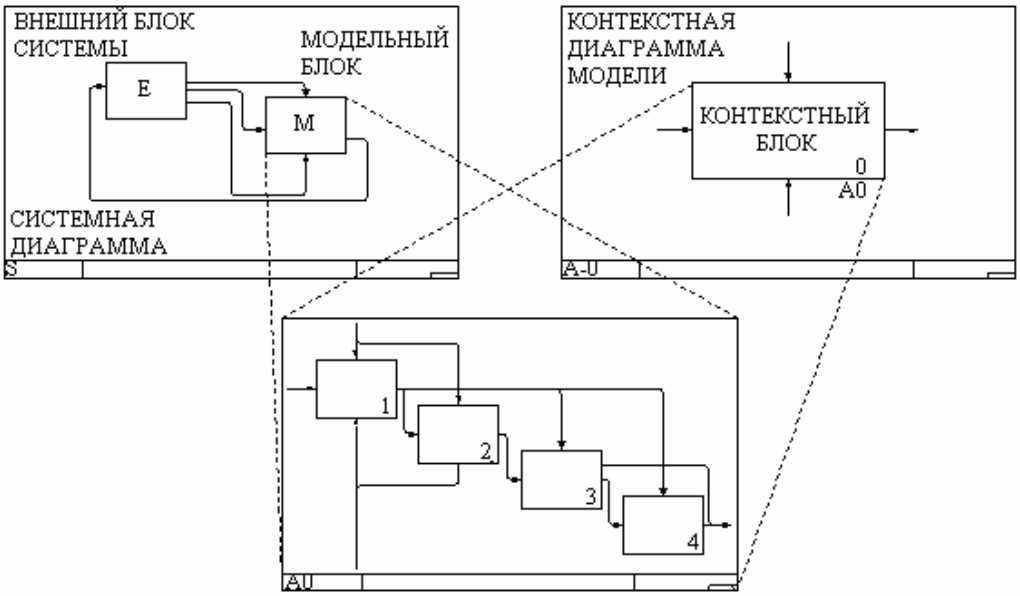
где I^D_{xy} – количество диаграмм в *y*-й модели *x*-й системы.

Модельный блок с его дугами определяет границу модели (бизнес-процесса), т.е. все, лежащее внутри него, является частью описываемой предметной области, а все, лежащее вне его, образует среду модели.

Каждая субмодель, как отмечалось, имеет собственную контекстную диаграмму, причем контекстная диаграмма нулевой субмодели фактически является контекстной диаграммой модели. Контекстная диаграмма модели (кроме системной модели) – это диаграмма, состоящая из одного контекстного блока и его дуг. Контекстный блок так же, как и модельный, изображает границу описываемой системы: все, лежащее внутри него, является частью описываемой системы, а все, лежащее вне его, образует среду системы. Здесь под системой понимается предметная область.

Контекстный блок модели – это копия соответствующего модельного блока с системной диаграммы. Оба блока имеют одни и те же входные и выходные дуги, однако из контекстной диаграммы модели нельзя определить, с какими моделями данная модель взаимодействует, как это видно из системной диаграммы.

Каждая *x*-я система моделей представлена на системной диаграмме D'_{x0}. Она обязательно содержит внешний блок системы B'_{x0}, представляющий ее внешнюю среду. Очевидно, что внешний блок системы не может иметь декомпозиции, управляющих дуг и дуг механизмов, поскольку он нужен только для того, чтобы быть источником и приемником дуг контекстных блоков моделей (входов и выходов), которые могут присутствовать на системной диаграмме. Структура системы *IDEF*0-моделей приведена на рисунке.



Структура системы IDEF0-моделей

Рассматриваемые выше множества имеют некоторую общность по группам. Так, множества атрибутов систем моделей, моделей, субмоделей и диаграмм содержат дополнительный атрибут – статус, принимающий значения: *Working* (рабочая версия), *Draft* (черновая версия), *Recommended* (рекомендовано к публикации) и *Publication* (публикация).

В заключение следует отметить, что методология *IDEF*0 используется для моделирования широкого круга требований и функций, а затем также и для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции.

Однако анализ и формализация средств *IDEF*0-методологии приводят к выводу, что для ее применения, например, при реструктуризации организаций требуются ее развитие с наложением ряда ограничений и разработка формальных алгоритмов модификации *IDEF*0-моделей, подробно рассматриваемых в [1, 2].

СПИСОК ЛИТЕРАТУРЫ

1. *Костров, А.В.* Уроки информационного менеджмента. Практикум: учеб. пособие / А.В. Костров, Д.В. Александров. – М.: Финансы и статистика, 2005. – 304 с. – ISBN 5-279-02573-9.

2. *Александров, Д.В.* Распределенные информационные системы. CASE-техноло-гии реинжиниринга: учеб. пособие / Д.В. Александров, А.В. Костров; Владимирский гос. ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2001. – 136 с. – ISBN 5-89368-241-6. ГИАБ

Коротко об авторе

Александров Д.В – докторант, кандидат технических наук, доцент, Владимирский государственный университет, г. Владимир.

Рецензент д-р техн. наук, проф., Заслуженный деятель науки РФ *А.В. Костров.*