

ФОРМИРОВАНИЕ КАРКАСА UML–ДИАГРАММЫ ДЕЯТЕЛЬНОСТИ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ УНИВЕРСАЛЬНЫХ ОБЪЕКТОВ МОДЕЛИ IDEF0

С.Н. Широбокова, Г.Н. Хубаев, Ю.В. Ткаченко, Е.В. Титаренко

ГОУ ВПО «Ростовский государственный экономический университет «РИНХ», г. Ростов-на-Дону; ГОУ ВПО «Южно-Российский государственный технический университет», г. Новочеркасск

Ключевые слова и фразы: автоматизация; бизнес-процессы; имитационное моделирование; конвертирование; IDEF0; UML.

Аннотация: Представлены формализованная модель и алгоритм построения каркаса UML-диаграммы деятельности, использующие универсальные объекты модели IDEF0, послужившие основой при разработке Автоматизированного конвертера диаграмм IDEF0 в UML-диаграммы деятельности. Описанная модель, алгоритм и правила преобразования, на которых он основан, являются оригинальными.

В [1,2] выдвинуто концептуальное положение о возможности и прикладной полезности конвертирования диаграмм IDEF0 в UML-диаграммы, предложен обобщенный алгоритм Автоматизированного конвертера диаграмм IDEF0 в UML-диаграммы с последующим экспортом данных в Конструктор [3] для генерации имитационной модели.

Известно, что унифицированный язык моделирования UML тесно интегрирован с используемыми методами разработки программных продуктов. Однако, как показали проведенные нами исследования, не менее успешно этот язык может использоваться для моделирования деловых процессов в различных предметных областях. Возможности UML позволяют моделировать процессы и потоки как активные классы и объекты, идентифицировать функции анализируемой системы, описывать поведение системы или ее части, задавать требования, связанные со временем и ограничениями по времени, указывать, где выполняется тот или иной компонент, определять его местонахождение. Используя «плавающие дорожки» на диаграммах деятельности, можно показать, какой объект отвечает за каждый вид деятельности. Кроме того, для оценки характеристик абсолютного большинства деловых процессов необходимо выполнять имитационное моделирование. А при использовании Конструктора [3] имитационная модель делового процесса генерируется автоматически по построенной UML-модели. Но поскольку диаграммы IDEF0 начали использоваться для моделирования деловых процессов задолго до появления языка UML, к настоящему времени в рамках методологии IDEF0 сформирована достаточно крупная база функциональных моделей. Принимая во внимание исключительно низкие трудозатраты на построение по визуальной UML-модели имитационной модели делового процесса, представляется целесообразным рассмотреть возможность конвертирования диаграмм IDEF0 в UML-диаграммы с последующей генерацией (в Конструкторе [3]) имитационной модели.

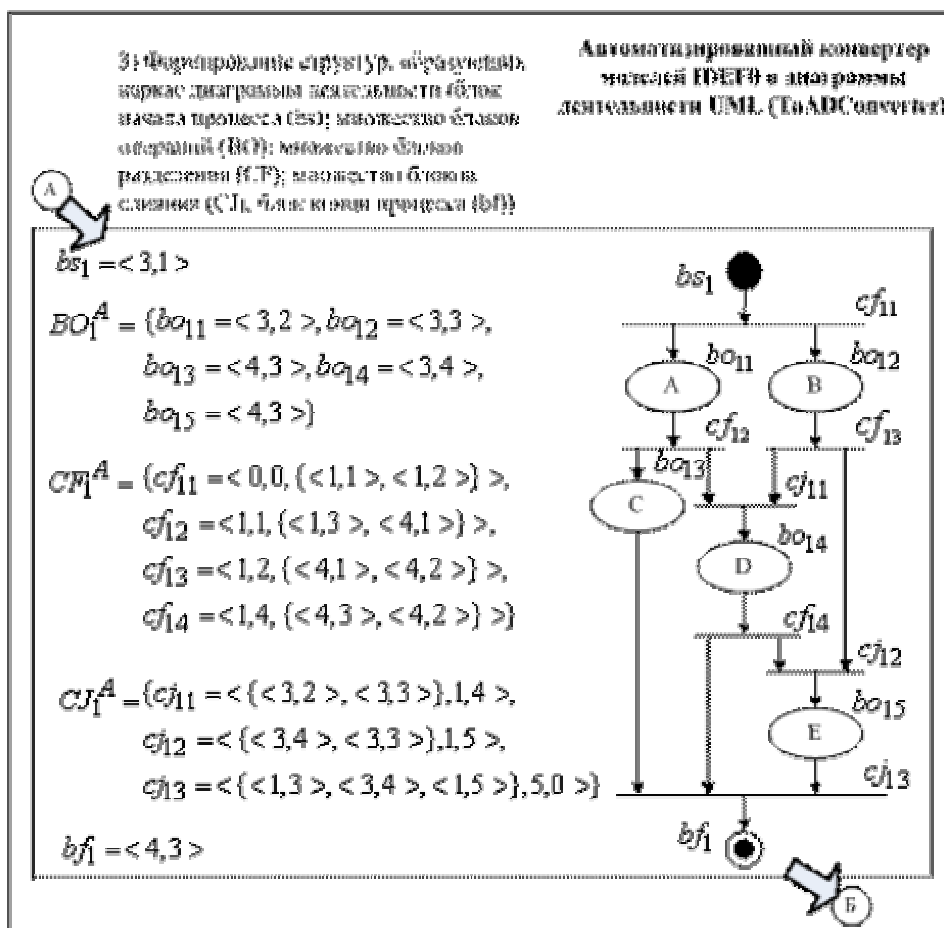


Рис. 2. Реализация конвертирования диаграммы IDEF0 в UML-диаграмму деятельности: этапы формирования каркаса UML-диаграммы деятельности

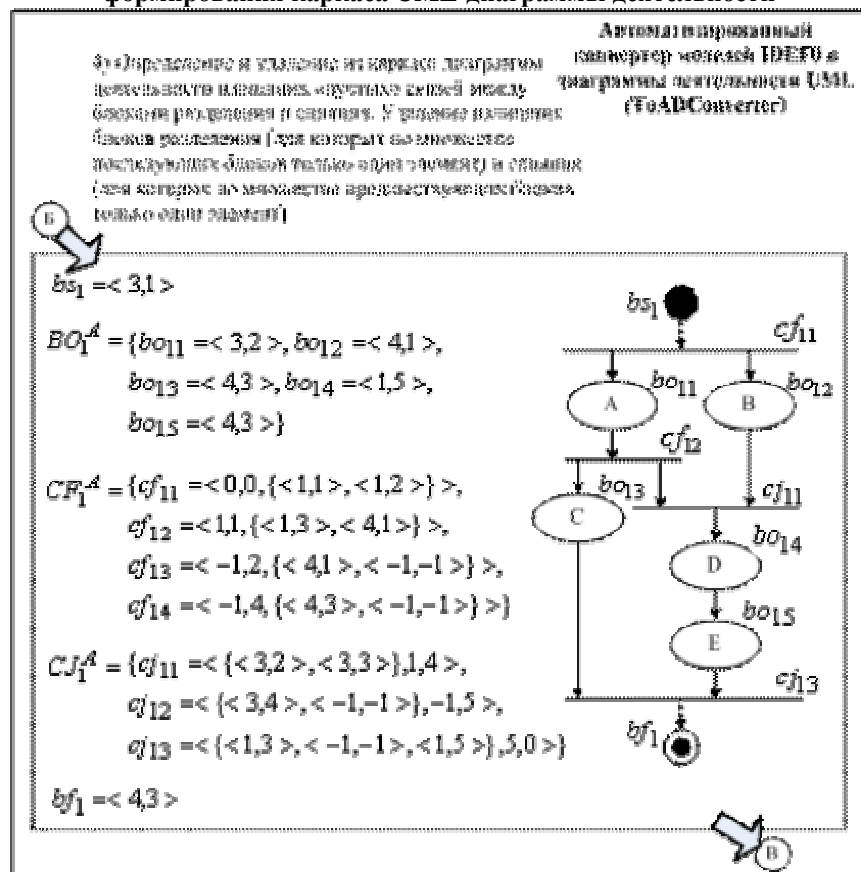


Рис. 3. Реализация конвертирования диаграммы IDEF0 в UML-диаграмму деятельности: этап удаления «пустых» связей из каркаса UML-диаграммы деятельности

В [4] описан алгоритм формирования универсальных объектов модели IDEF0, экспортируемой из ALLFusion Process Modeler 7 (BPWin). Ниже представлены структуры, используемые на этапе формирования каркаса диаграммы деятельности (рис. 2) по данным объекта TDiagram модели IDEF0 [4].

Общие переменные модели IDEF0:

Q – число диаграмм в модели (учитывая контекстную диаграмму);

$m = \overline{2, Q}$ – номер диаграммы декомпозиции в модели IDEF0;

qb_m – число блоков (работ) на m -ой диаграмме декомпозиции модели IDEF0;

qar_m – число стрелок на m -ой диаграмме декомпозиции IDEF0.

Формализованное описание структур, полученных на этапе формирования универсальных объектов модели IDEF0:

$B_m^{DD} = \{b_t^m\}$, $t = \overline{1, qb_m}$ – множество блоков (работ) m -ой диаграммы декомпозиции модели IDEF0.

Элемент множества b_t^m описан универсальным объектом TBlock;

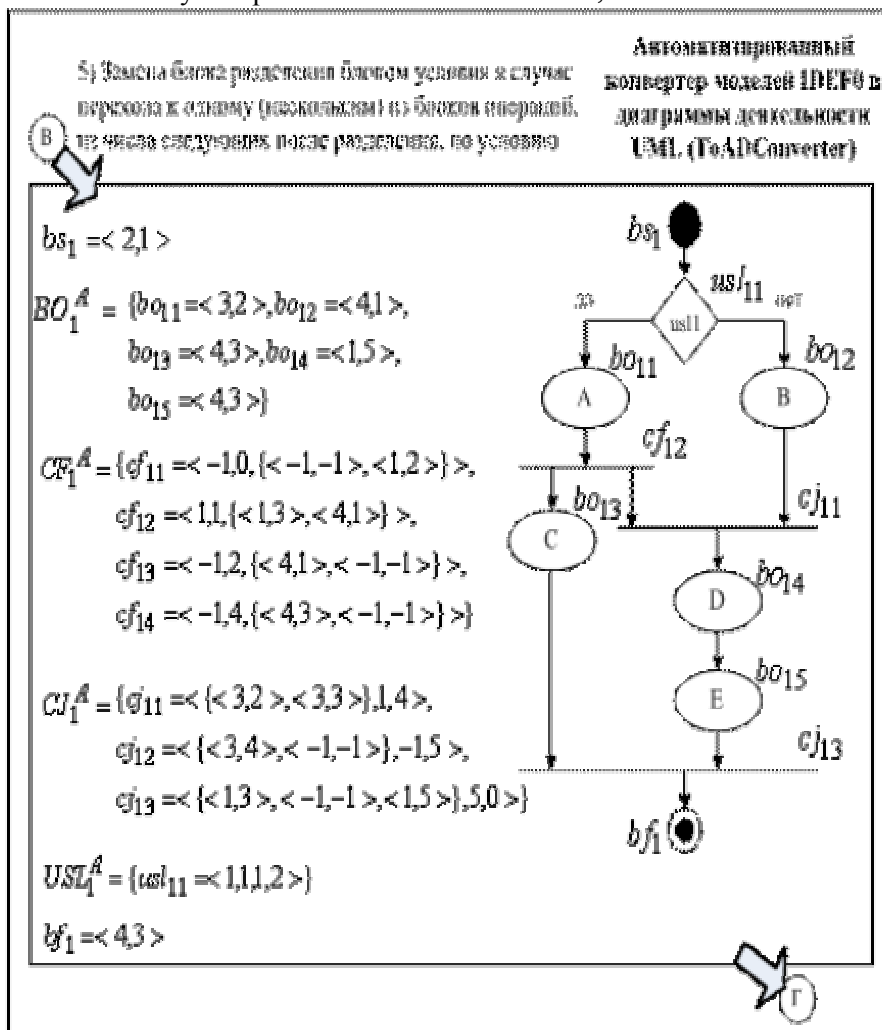


Рис. 4. Реализация конвертирования диаграммы IDEF0 в UML-диаграмму деятельности: вставка в каркас UML-диаграммы деятельности блоков условий

$AR_m^{DD} = \{ar_k^m\}$, $k = \overline{1, qar_m}$ – множество стрелок m -ой диаграммы декомпозиции модели IDEF0.

Элемент множества ar_k^m описан универсальным объектом TArrow;

$FIRST_m = \{first_t^m\}$, $t = \overline{1, qb_m}$ – массив, в котором отмечены первые работы процесса, описанного диаграммой декомпозиции IDEF0. Если работа t является первой в процессе, то элемент $first_t^m = 1$;

$LOBR_m = |lobr_k^m|$, $k = \overline{1, qar_m}$ – массив, в котором отмечаются стрелки, определяющие возврат по циклу.

Если стрелка k определяет возврат по циклу, то элемент $lobr_k^m = 1$;

$POZ_m = |poz_{kt}^m|$, $k = \overline{1, qar_m}$, $t = \overline{1, qb_m}$ – матрица взаимосвязей блоков m -ой диаграммы декомпозиции модели IDEF0. Элемент матрицы принимает значение $poz_{kt}^m = 1$, если стрелка ar_k^m является выходом из блока b_t^m ;

$poz_{kt}^m = 2$ – если стрелка ar_k^m является входом типа Input в блок b_t^m ; $poz_{kt}^m = 3$ – если стрелка ar_k^m является входом типа Control в блок b_t^m ; $poz_{kt}^m = 4$ – если стрелка ar_k^m является входом типа Mechanism в блок b_t^m ; $poz_{kt}^m = 0$ – если стрелка не связана с блоком; $poz_{kt}^m = 5$ – стрелка определяет цикл по одной работе t по входу типа Input; $poz_{kt}^m = 6$ – стрелка определяет цикл по одной работе t по входу типа Control.

Цифровое обозначение типов блоков на диаграмме деятельности UML (указаны блоки, используемые на этапе формирования каркаса диаграммы):

- 0 – блок начала процесса bs_m на диаграмме деятельности UML;
- 1 – блок операции на диаграмме деятельности UML (элемент множества BO_m^A);
- 3 – блок разделения на диаграмме деятельности UML (элемент множества CF_m^A);
- 4 – блок слияния на диаграмме деятельности UML (элемент множества CJ_m^A);
- 5 – блок конца процесса bf_m на диаграмме деятельности UML.

Формализованное описание блоков, используемых на этапе формирования каркаса диаграммы деятельности UML:

$bs_m = \langle ntype_m, nmb_m \rangle$ – блок начала диаграммы модели UML, где $ntype_m$ – тип следующего блока после начального состояния; nmb_m – номер следующего блока после начального состояния в соответствии с типом блоков $ntype_m$;

$BO_m^A = |bo_{mt}|$, $t = \overline{1, qb_m}$ – множество блоков операций на диаграмме деятельности модели UML. Блок операции описан следующей структурой: $bo_{mt} = \langle type_{mt}, nl_{mt} \rangle$, где $type_{mt}$ – тип следующего за блоком операции блока, nl_{mt} – номер следующего блока в соответствии с нумерацией блоков типа $type_{mt}$;

$CF_m^A = |cf_{ml}|$, $l = \overline{1, NL_m}$ – множество блоков разделения на диаграмме деятельности модели UML, где NL_m – число блоков разделения, которое определяется динамически, по мере необходимости ввести данный элемент. Блок разделения характеризуется следующей структурой: $cf_{ml} = \langle tprev_{ml}^{cf}, nprev_{ml}^{cf}, ENEXT_{ml}^{cf} \rangle$, где $tprev_{ml}^{cf}$ – тип блока, предшествующего блоку разделения, $nprev_{ml}^{cf}$ – номер блока, предшествующего блоку разделения, в соответствии с нумерацией блоков типа $tprev_{ml}^{cf}$ ($nprev_{ml}^{cf} = 0$, если $tprev_{ml}^{cf} = 0$), $ENEXT_{ml}^{cf} = \{bnext_{ml}^{cf}\}$, $i = \overline{1, NEXT_{ml}}$ – множество следующих блоков после элемента разделения.

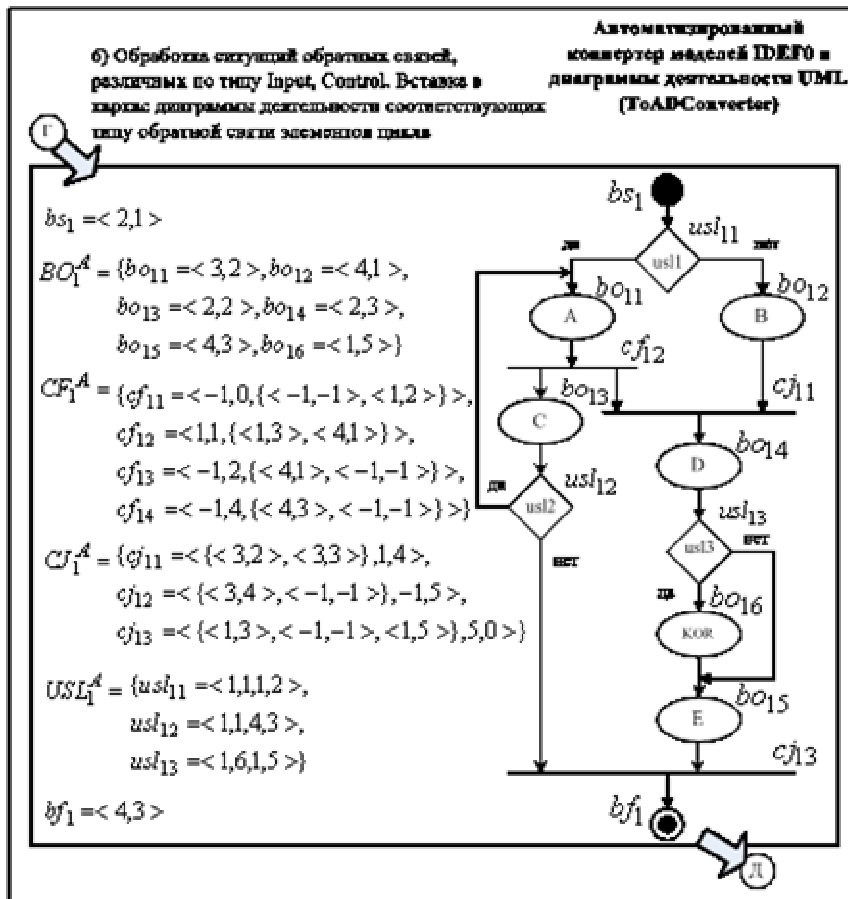


Рис. 5. Реализация конвертирования диаграммы IDEF0 в UML-диаграмму деятельности: вставка элементов цикла в каркас диаграммы деятельности UML

Блок слияния характеризуется следующей структурой:

$$qj_{ms} = \langle BPREV_{ms}^{qj}, tnext_{ms}^{qj}, nnext_{ms}^{qj} \rangle$$

, где $BPREV_{ms}^{qj} = \{bprev_{msj}^{qj}\}$,

$j = 1, PREV_{ms}$ – множество предшествующих блоку слияния блоков, где $PREV_{ms}$ – число предшествующих блоку слияния, $tnext_{ms}^{qj}$ – тип последующего блока после блока слияния, $nnext_{ms}^{qj}$ – номер последующего блока после блока слияния в соответствии с нумерацией блоков типа

$tnext_{ms}^{qj}$ ($nnext_{ms}^{qj} = 0$, если $tnext_{ms}^{qj} = 5$). Элементом множества $BPREV_{ms}^{qj}$ является $bprev_{msj}^{qj} = \langle tprev_{msj}^{qj}, nprev_{msj}^{qj} \rangle$, $j = 1, PREV_{ms}$, где $tprev_{msj}^{qj}$ – тип предшествующего блока, а параметр $nprev_{msj}^{qj}$ – номер предшествующего блока в

соответствии с нумерацией блоков типа $tprev_{msj}^{qj}$;

$bf_m = \langle ptype_m, pnib_m \rangle$ – блок конца диаграммы модели UML, где $ptype_m$ – тип предшествующего конечному состоянию блока; $pnib_m$ – номер предшествующего конечному состоянию блока в соответствии с нумерацией блоков типа $ptype_m$.

На рис. 1–6 представлены схемы, иллюстрирующие основные этапы конвертирования моделей, реализованные в Автоматизированном конвертере.

Первым шагом построения диаграммы деятельности UML является формирование каркаса (рис. 2), который в дальнейшем оптимизируется и дополняется в зависимости от особенностей диаграммы декомпозиции IDEF0. Формализованное описание алгоритма построения каркаса представлено ниже.

Шаг 1. Формирование множества CF_m^A .

Условие 1. Проверка на количество первых работ.

Введем некоторый счетчик $nfirst$ первых работ на диаграмме IDEF0. Если в массиве первых работ $FIRST_m$ найдено $nfirst > 1$ число элементов $first_t^n = 1$, $t \in \overline{1, qb_m}$, следовательно, первых работ процесса несколько.

Действие 1. Формирование элемента разделения после начального состояния процесса.

Если для m -ой диаграммы декомпозиции модели IDEF0 выполняется Условие 1 и известны номера столбцов матрицы POZ_m , которые описывают первые работы: $t_1, t_2, \dots, t_{nfirst}$, то следующим после начального элемента диаграммы деятельности UML следует поставить блок разделения.

Увеличивается динамический счетчик добавленных элементов разделения $NL_m = NL_m + 1$.

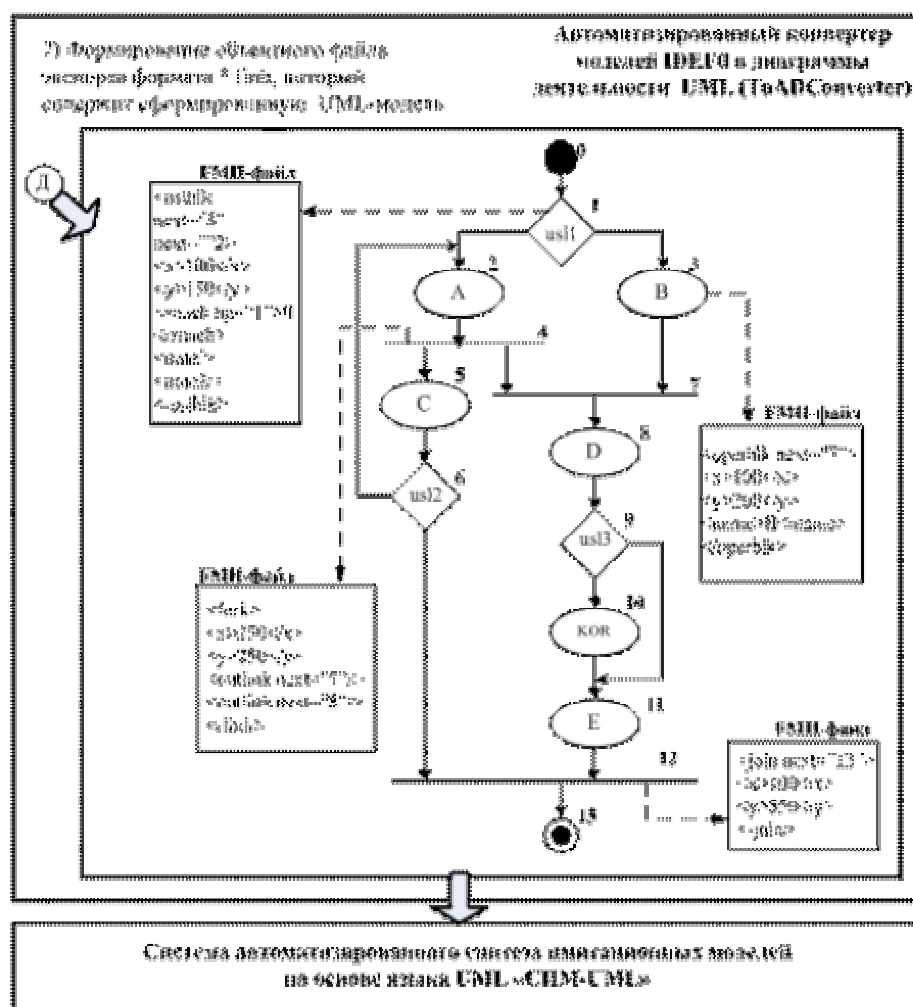


Рис. 6. Реализация конвертирования диаграммы IDEF0 в UML-диаграмму деятельности: этап формирования объектного файла экспорта

Для добавляемого во множество CF_m^A блока разделения cf_{ml} : $l = NL_m$, число последующих блоков $NEXT_{ml} = nfirst$. Устанавливаются параметры: $tprev_{ml}^{df} = 0$, $nprev_{ml}^{df} = 0$, $BNEXT_{ml}^{df} = \{bnext_{ml1}, bnext_{ml2}, \dots, bnext_{mlnfirst}\}$, где $bnext_{ml1} = \langle 1, t_1 \rangle$, $bnext_{ml2} = \langle 1, t_2 \rangle, \dots$, $bnext_{mlnfirst} = \langle 1, t_{nfirst} \rangle$.

Действие 2. Поиск работ, после которых необходимо поставить элемент разделения.

2.1. Введем некоторый счетчик выходов out из блока на диаграмме IDEF0. Пусть для m -ой диаграммы декомпозиции IDEF0 проверяется столбец $t_r \in \overline{1, qb_m}$ матрицы POZ_m на наличие элементов выхода $poz_{kr}^m = 1$ из блока t_r . Пусть таких элементов найдено $out > 1$, следовательно, работа имеет несколько выходов.

Для каждого $poz_{kr}^m = 1$ из числа out проверяется Условие 2а. В случае нарушения Условия 2а для элемента $poz_{kr}^m = 1$ выполняется действие: $out = out - 1$.



Условие 2а. Проверка на циклический выход.

Условие 2б. Проверка необходимости блока разделения.

Если, при выполнении п. 2.1 Действия 2, в результате проверки по столбцу $t_{\gamma} \in \overline{1, qh_m}$ матрицы POZ_m для каждого элемента выхода $POZ_{m, t_{\gamma}}^{out} = 1$ Условия 2а число этих элементов выхода $out = 1$, значит, блок разделения не ставится после работы на диаграмме деятельности UML, т.к. переход, не являющийся циклическим, из работы всего один.

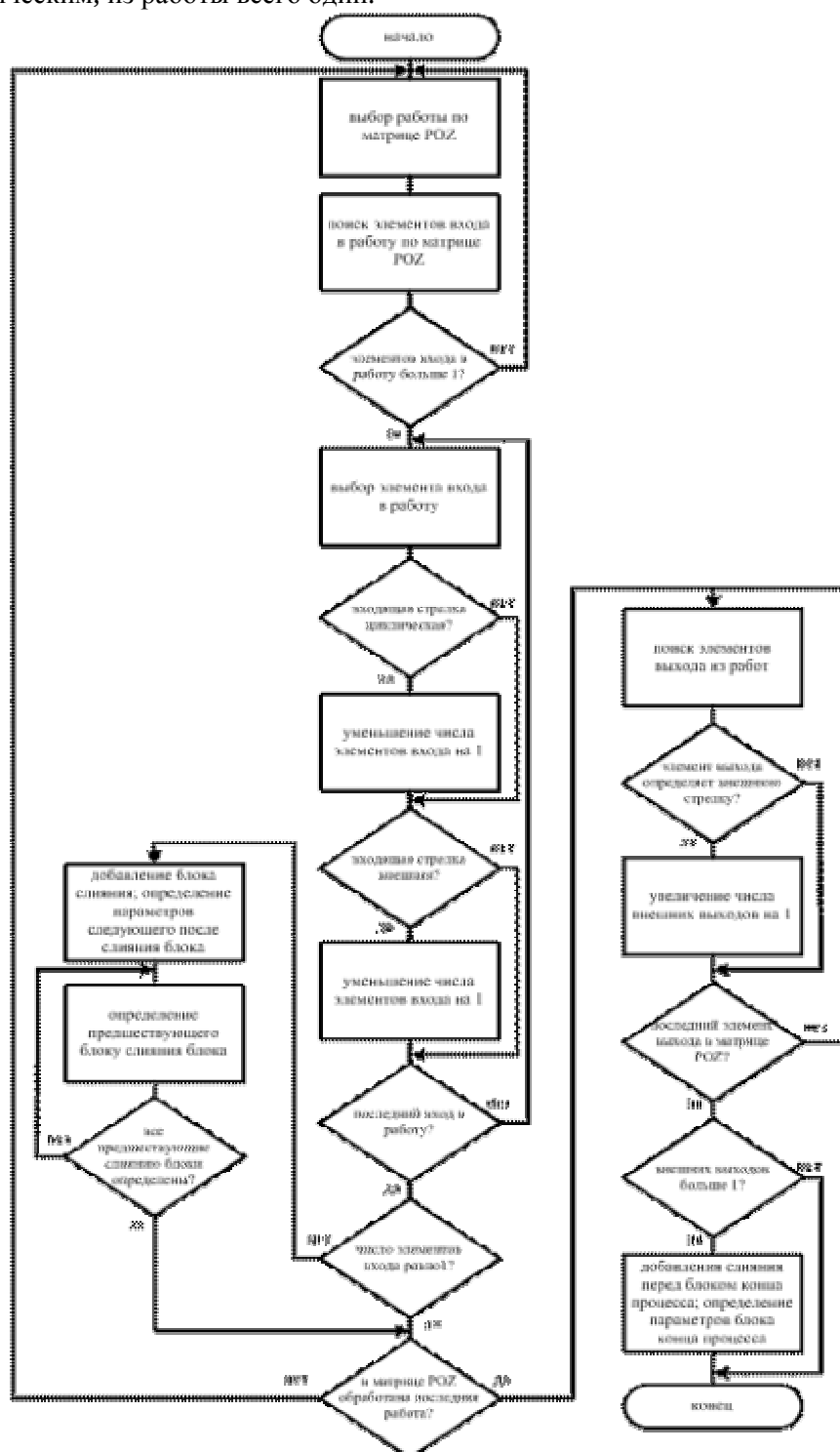


Рис. 8. Алгоритм формирования множества блоков слияния на диаграмме деятельности UML

Действие 3. Определение параметров элемента разделения.

3.1. Пусть для столбца $t_{\gamma} \in \overline{1, qh_m}$ матрицы POZ_m не выполняется Условие 2б, тогда необходимо на диаграмме деятельности UML после блока с номером t_{γ} поставить элемент разделения. Увеличивается динамический счетчик добавленных элементов разделения $NL_m = NL_m + 1$. Для добавляемого блока

разделения $cf_{ml} : l=NL_m$, число последующих блоков $NEXT_{m3} = out$. Устанавливаются параметры $tprev_{mj}^{cf} = 1$, $mprev_{mj}^{cf} = f$. Далее переход к п.п. 3.2.

3.2. Формируется множество следующих после блока разделения блоков:
 $BPREV_{m3}^{cf} = \{bprev_{m31}^{cf}, bprev_{m32}^{cf}, \dots, bprev_{m3in}^{cf}\}$.

Блок-схема алгоритма формирования множества блоков разделения CF_m^A на диаграмме деятельности UML изображена на рис.7.



Рис. 9. Алгоритм формирования ссылок между блоками разделения и слияния на диаграмме деятельности UML

диаграммы деятельности. Описанная модель, алгоритм и правила преобразования, на которых он основан, являются оригинальными.

Список литературы

1. Хубаев, Г.Н. Автоматизированный конвертер моделей IDEF0 в UML-диаграммы: концептуальная идея / Г.Н. Хубаев, С.Н. Широбокова // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей VIII Всероссийской науч.-техн. конф., г. Пенза, 19–20 нояб. 2008. – Пенза : Приволжский Дом знаний, 2008. – С. 102–108.
2. Хубаев, Г.Н. Конвертирование диаграмм IDEF0 в UML-диаграммы: концепция и правила преобразования / Г.Н. Хубаев, С.Н. Широбокова // Проблемы экономики, 2008. – №6. – С. 139–152.
3. Хубаев, Г.Н. Система автоматизированного синтеза имитационных моделей на основе языка UML «СИМ-UML» / Г.Н. Хубаев, С.М. Щербаков, Ю.А. Рванцев // Свидетельство об официальной регистрации программы для ЭВМ. – № 2009610414. – М. : РОСПАТЕНТ, 2009.
4. Широбокова, С.Н. Алгоритм модуля синтаксического разбора файла экспорта диаграмм IDEF0 для создания универсальных объектов / С.Н. Широбокова, Е.В. Титаренко, Ю.В. Ткаченко // Компьютерные технологии в науке, производстве, социальных и экономических процессах: Материалы IX Международной науч.-практ. конф., г.Новочеркасск, 17 нояб. 2008. – Юж.-Рос. гос. техн. ун-т (НПИ). – Новочеркасск: ЮРГТУ, 2008. – С. 112–119.

Formation of Carcass UML Diagram Based on Application of IDEF0 Model Universal Objects

S.N. Shirobokova, G.N. Khubaev, Y.V. Tkachenko, E.V. Titarenko

Rostov State Economic University «RINKH», Rostov-on-Don

South-Russian State Technical University, Novocherkassk

Key words and phrases: automation; business processes; simulation; converting; IDEF0; UML.

Abstract: The paper presents the formalized model and algorithm for the formation of the carcass of UML-activity diagram, using the universal object of IDEF0 model, which is served as the basis for the development of automated IDEF0 diagrams into UML-activity diagram converter. The described model, the algorithm and the transformation rules on which it is based, are original.