УДК 65.012.123

В.Ю. Карпычев

ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ (IDEF0) КАК МЕТОД ИССЛЕДОВАНИЯ БЛОКЧЕЙН-ТЕХНОЛОГИИ

Нижегородский государственный технический университет им. Р.Е. Алексеева

Функциональное моделирование на основе стандарта IDEF0 – создание графических моделей любой предметной деятельности, включающих иерархическое описание процессов, операций, ресурсов (информации), инструментария, исполнителей, управления и связей между ними. В статье предложена функциональная модель управления активами различной природы на основе блокчейн-технологии. Модель не только специфицирует основные функциональные блоки исследуемой БЧ-системы, но и наглядно представляет логику ее работы путем последовательной декомпозиции целевой функции до уровня предметных операций. В модели интегрированы понятия предметной области (экономической деятельности) и информационных технологий. Использованный в модели IDEF0-язык позволяет обеспечить понятийный интерфейс предметных и IT-специалистов.

Ключевые слова: управление активами, функциональное моделирование, блокчейн-технологии.

Введение

В настоящее время большое внимание ученых и специалистов в различных областях привлекает блокчейн (БЧ) технология. Известно много определений данного термина. В контексте данной статьи будем рассматривать *блокчейн* как «многофункциональную и многоуровневую информационную технологию, предназначенную для надежного учета различных активов» [1]. На портале «e-library» размещены сведения о сотнях русскоязычных публикаций по БЧ-тематике экономической, правовой, технической и управленческой направленности. Однако среди этого обилия работ только некоторые отличаются глубиной, системностью и доступностью изложения теории БЧ-технологии. В статье предпринята попытка интерпретировать БЧ-технологию в моделях методологии/стандарта SADT/IDEF0.

Для создания модели БЧ-технологии использована книга Д. Дрешера [2] как наиболее отвечающая, по мнению автора, указанным требованиям. Методология IDEF0 изложена во многих работах, в качестве первоисточника рекомендуем [3]. Стандарт IDEF0 принят в США [4] и рекомендован Госстандартом России для исследования структуры, параметров и характеристик производственных и организационно-экономических систем [5, 6]. Главными компонентами IDEF0-модели являются диаграммы, графически представляющие структуру функций предметной области, а также информации и объектов, связывающих эти функции. Функции и их интерфейсы представлены как блоки и дуги соответственно. При IDEF0-моделировании производится декомпозиция предметной контекстной (целевой) функции на согласованные и непротиворечивые функции следующего уровня детализации. Предметная функция любого уровня, начиная с контекстной, преобразует входной поток сущностей – физических объектов или информации в выходной. Будем называть такие потоки **Входом** и **Выходом** функции. Для активации функций необходимы **Управление** (управленческое воздействие), которое определяет условия выполнения функции, и **Механизмы** – средства, непосредственно реализующие функцию. Для поддержки моделирования используются CASE (Computer Aided Software Engineering) средства.

Характеристика предметной области

В качестве предметной области моделирования будем рассматривать экономический оборот активов различной природы. Основная процедура оборота – *обмен* (передача) активов.

В современной экономике значительная часть активов существует в форме документированной цифровой информации, в том числе сведений о количестве актива. Тогда оборот актива – не что иное, как уменьшение количества актива у одного субъекта отношений и приращение его – у другого. При наличии волеизъявления субъектов и необходимых организационно-правовых условий оборот актива можно рассматривать как бизнес-процесс (БП) «**Управлять правом владения активом**», A_0. (например, передавать цифровые денежные средства).

IDEF0-модель

Построение IDEF0-модели БП начинается с контекстной диаграммы, A_0. описывающей в общем виде работу системы в предметной области (рис. 1а).

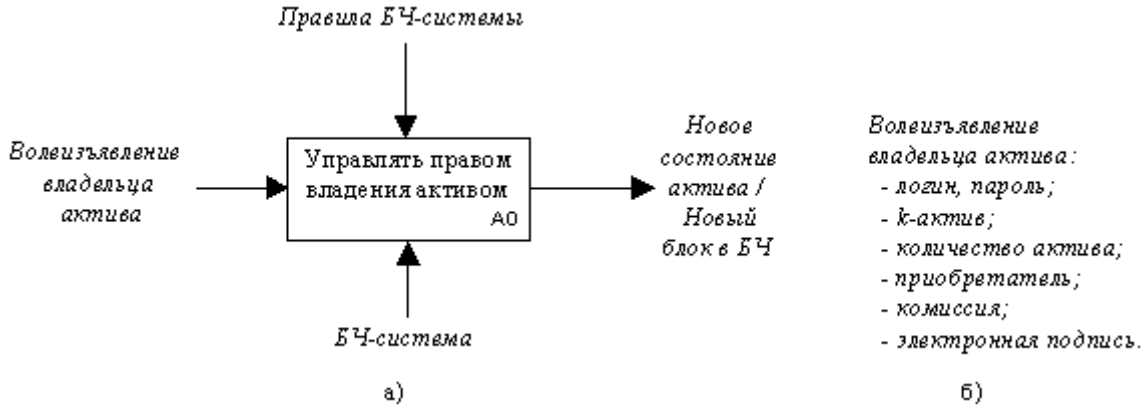

Рис. 1. Контекстная диаграмма БП «Управлять правом владения активом», A_0

В качестве **Входа** будем рассматривать *Волеизъявление владельца актива* на передачу части/всего актива любому приобретателю, отвечающему рассматриваемым ниже требованиям. **Выходом** диаграммы является *Новое состояние <количество> актива* у субъектов экономического оборота. **Управление** сформулируем в общем виде: *Правила работы БЧ-системы.* **Механизм:** техническим средством поддержки БП A_0, является БЧ-система – автоматизированная система, основные компоненты которой – вычислительные средства (ВС), операторы ВС и P2P-сеть, реализованная в Интернете. Объединим понятие ВС и его оператора термином «узел» и будем рассматривать, соответственно, множество узлов. Некоторые узлы (операторы ВС) являются владельцами/приобретателями актива $N = \{1, n\}$ и используют БЧ-систему по назначению. Остальные – M-узлы (майнеры) – обеспечивают функционирование БЧ-системы (выполняют технологические функции). Множества M и N пересекаются.

Внутренняя организация и функционал БП A_0, использующего БЧ-систему, имеют оригинальные решения. Кратко рассмотрим их. Любая система поддержки управления правом владения активом должна документировать это право, то есть устанавливать связь владельца и актива. В БЧ-системе документирование права владения осуществляется ведением реестра хронологии транзакций, фиксирующим все факты передачи права владения. Каждая передача права владения описывается данными транзакции: владелец, передаваемый актив, приобретатель актива, количество и время передачи актива. В технических терминах транзакция содержит данные об учетных записях, передающей и принимающей право владения конкретным активом. Данные транзакций хранятся в порядке выполнения в структуре данных БЧ. Включение транзакций в БЧ меняет данные владельцев актива, определяет текущего владельца и подтверждает действительность транзакций. Описанная концепция организации БЧ-системы позволяет переформулировать **Выход** процесса A_0 в терминах БЧ-технологии: *Новый блок в БЧ.*

На рис. 2 представлен первый уровень декомпозиции БП A_0, который можно рассматривать как последовательное достижение двух предметных целей: «Подтверждение права владения активом» и «Передача права владения активом».

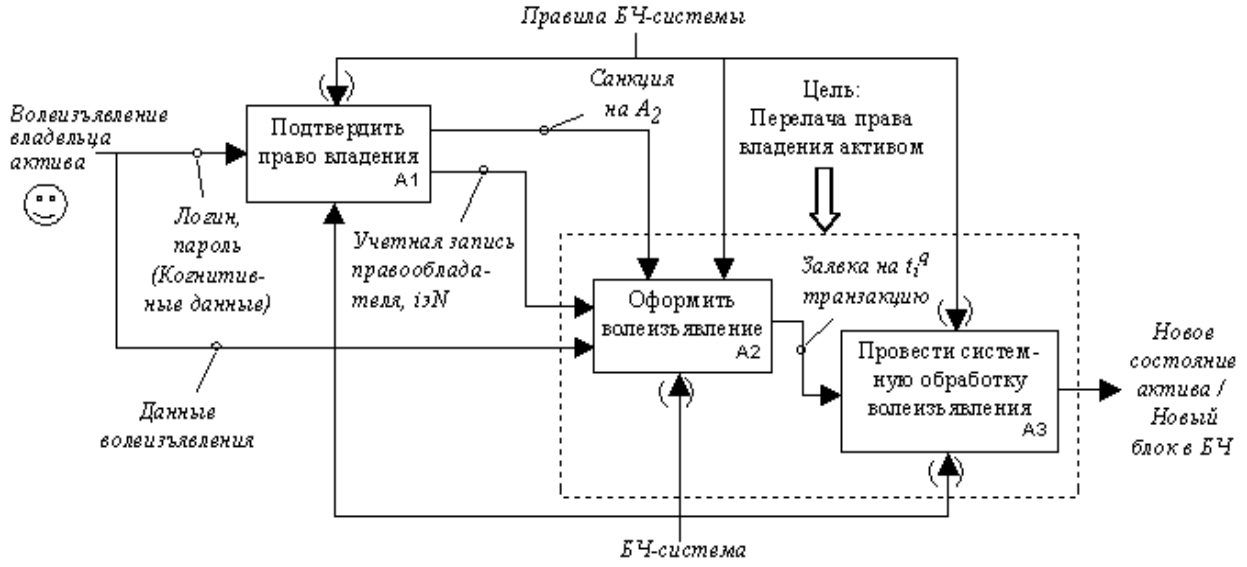

Рис. 2. Декомпозиция БП «Управлять правом владения активом», A_0

Технически право владения активом может быть подтверждено путем предоставления доступа к активу. Тогда в модели процесс A_1 можно именовать «**Получить доступ в БЧ-систему**» и интерпретировать его назначение как исключение несанкционированного доступа к активу. В БЧ-системе передача права владения активом предусматривает выполнение как предметно значимых действий (процесс «**Оформить волеизъявление**», A_2), так и исключительно технологических процедур (процесс «**Провести системную обработку волеизъявления**», A_3). Обработка волеизъявления обеспечивается структурно-параметрическими и функциональными решениями БЧ-системы. Семантика процесса A_1: каждая транзакция должна содержать информацию, документирующую волеизъявление владельца актива на его передачу. Поэтому доступ в БЧ-систему для выполнения транзакции разрешен только владельцу актива, или, в IT-терминах, владельцу соответствующей учетной записи.

Вход процесса A_1. Волеизъявление владельца является сложной информационной сущностью (рис. 1б). В контексте процесса A_1 используются реквизиты этой сущности *логин* и *пароль* (когнитивные данные). Также для выполнения процесса необходимы данные *хранилища данных (ХД) учетных записей* БЧ-системы. Функционал процесса A_1 включает три стандартные функции управления доступом к учетной записи БЧ-системы: «Идентифицировать владельца» (A_{11}), «Аутентифицировать владельца» (A_{12}) и «Авторизировать <владельца> (Обеспечить доступ к активу)», (A_{13}), на рис. 3.

Процесс A_1 (A_{13}) имеет два **Выхода.**

1. Сущность *Санкция (разрешение функции) на A_2* на проведение операции с активами владельца.
2. Сущность *Учетная запись владельца*, поступающая на вход функции A_{214}.

Управление процессом A_{11} осуществляется в соответствии с *Порядком доступа в БЧ-систему*, который входит в *Правила БЧ-системы*. **Механизмы:** доступ в БЧ-систему производится с ВС любого *N*-узла сети. Для конкретизации дальнейшего описания работы БЧ-системы будем считать, что доступ происходит с *i*-узла ($i \ni N$). Примечание. Для удобства чтения некоторых диаграмм элементы **Управление** и **Механизмы** оформлены только текстовыми надписями (без соответствующих дуг). Также для повышения информативности на диаграммах приведено несколько графических примитивов, не предусмотренных нотацией

IDEF0. Пользовательская цель «Передача права владения активом» достигается последовательным выполнением процессов A_2 и A_3 (рис. 2) и с предметной точки зрения означает изменение текущего состояния прав владения активом.

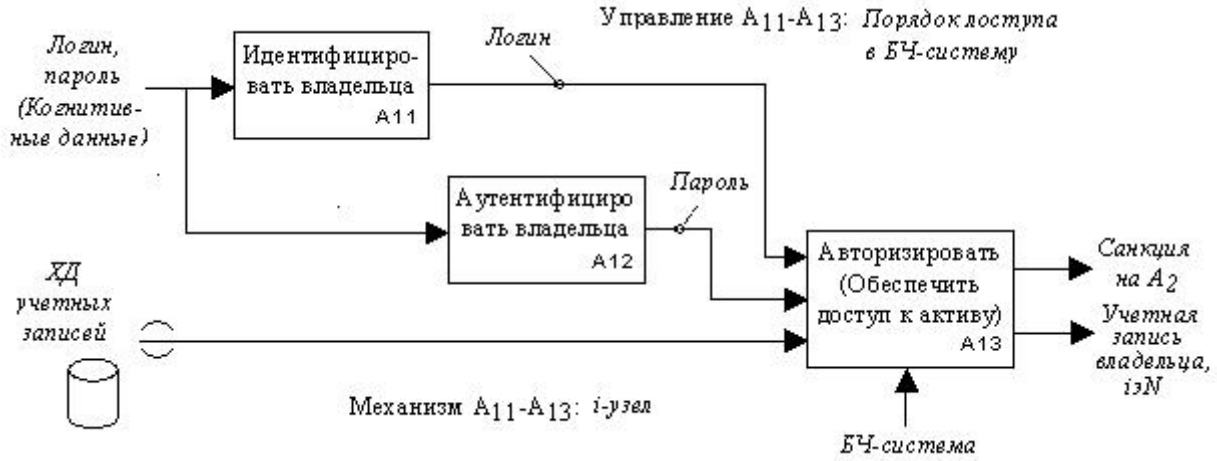

Рис. 3. Декомпозиция процесса «Получить доступ в БЧ-систему», A_1

Процесс «**Оформить волеизъявление**» A_2 состоит из процессов A_{21} и A_{22}, (диаграмма декомпозиции A_2 не приводится).

Семантика процесса «**Сформировать проект заявки на транзакцию**» A_{21} заключается в последовательном формировании владельцем актива данных планируемой транзакции. Термин «проект заявки…» использован для формализованного отражения намерений владельца на передачу актива.

На **Вход** процесса поступают когнитивные данные *i*-владельца: *передаваемый k-актив* ($k \in K$), при наличии у владельца нескольких активов *K*; *размер передаваемого актива и комиссии*, которая может быть уплачена за выполнение транзакции, данные *j-приобретателя* актива ($j \in N$). Также необходимы системные данные *хранилища данных учетных записей* и *текущее время.* Декомпозиция процесса A_{21} приведена на рис. 4.

На диаграмме отражены следующие функции.

- «Выбрать актив», A_{211} при работе БЧ-системы с несколькими активами;
- «Определить размер актива и комиссии», A_{212}, если предусмотрено возмездное выполнение транзакции;
- «Определить приобретателя актива» (учетная запись $j \in N$), A_{213};
- «Собрать проект заявки» на транзакцию», A_{214}.

Выход процесса A_{21} (A_{214}) – *проект заявки на транзакцию*, которая содержит следующие реквизиты: идентификатор учетной записи, передающей право владения *k*-активом; идентификатор учетной записи, принимающей право владения *k*-активом; количество передаваемого актива; время, выполнения транзакции; комиссия, предлагаемая БЧ-системе (майнеру) за проведение транзакции.

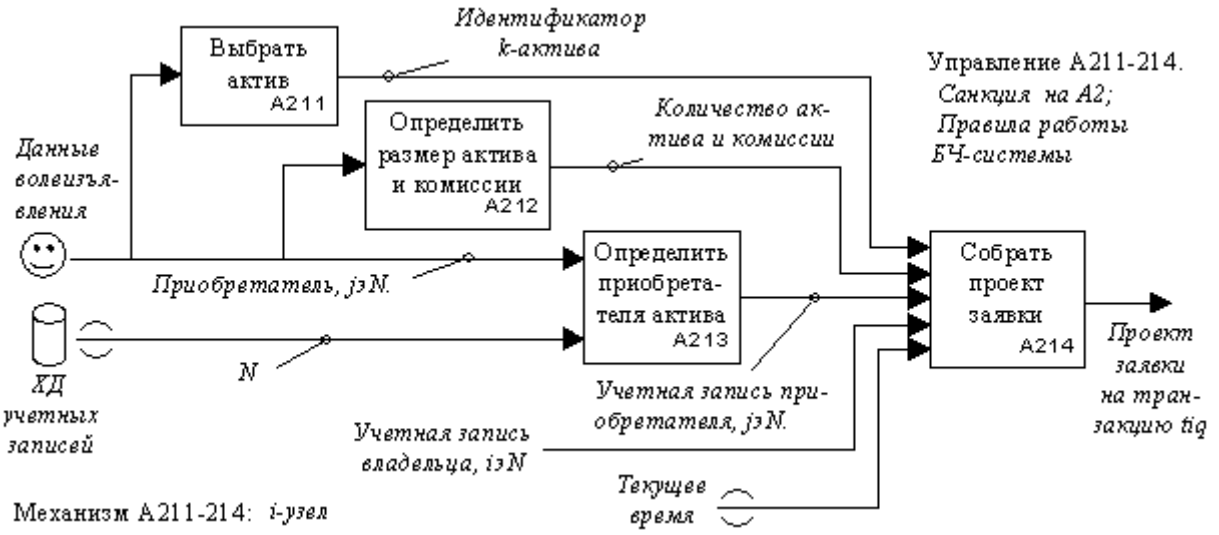

Рис. 4. Декомпозиция процесса «Сформировать проект заявки на транзакцию», A_{21}

Будем полагать, что *i*-узел формирует проект заявки на *q*-транзакцию $t_i^q \in T_i$, где T_i – множество транзакций *i*-узла в БЧ-системе. Для актуализации процесса A_{21} необходимы **Управление** – санкция процесса A_{13} (авторизация *i*-го владельца) и **Механизм**: ВС *i*-узла. Процесс «**Актуализировать заявку на транзакцию»,** A_{22} является комплексным и включает в себя предметные и технологические процедуры (рис. 5).

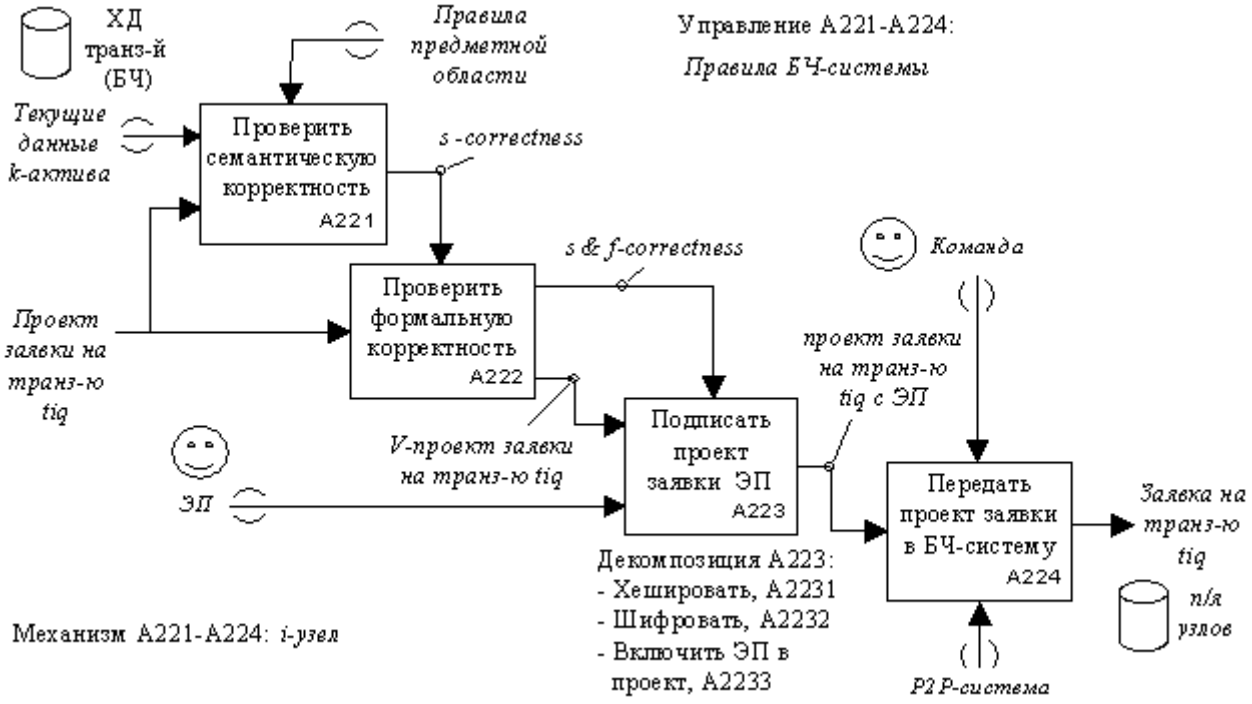

Рис. 5. Декомпозиция процесса «Актуализировать заявку на транзакцию», A_{22}

Семантика процесса A_{22} заключается в верификации (проверке корректности) и валидации сформированного *проекта заявки на транзакцию* t_i^q. Под *семантической корректностью* (*semantic correctness*) будем понимать соответствие данных транзакции правилам предметной области, например:

- количество заявляемого владельцем для передачи актива не должно превышать количество этого актива, находящегося во владении, или предельного значения для одной транзакции;
- количество транзакций владельца актива не должно превышать определенного значения в единицу времени;
- общее количество актива, передаваемого за определенный интервал времени, не должно превышать установленного значения;
- время владения активом не должно быть менее некоторого значения и др.

Формальная корректность (*formal correctness*) – корректный формат описания транзакции – устанавливается *правилами БЧ-системы*.

Валидация является результатом применения электронной подписи (ЭП), гарантирующей подлинность, невозможность отказа от авторства транзакции (неотрицаемость), неизменность подписанных данных (целостность).

Входами процесса A_{22} (A_{221}) являются *проект заявки на транзакцию* t_i^q и *текущие данные k-актива.* Отметим, что термин *текущие данные* не является синонимом термину *баланс,* поскольку БЧ реализует транзакционный учет данных (не ведет баланс) актива [6]. Поэтому на вход A_{22} БЧ-система выдает БЧ – хронологию транзакций *k-актива,* данные которых используются в семантической проверке.

Верификация *проекта заявки на транзакцию* включает функции «Проверить семантическую корректность», A_{221} и «Проверить формальную корректность», A_{222} <проекта заявки на транзакцию>.

Выходами функций верификации является верифицированный *V-проект заявки на транзакцию* и санкция *s & f-correctness* на подписание *V-проекта* ЭП *i*-владельца актива (**Управление** на A_{222}).

Приведенные выше *Правила предметной области* можно рассматривать как **Управление** функцией A_{221}, а *Правила БЧ-системы* в части требований к формату транзакции – как **Управление** функцией A_{222}.

Процесс A_{22} включает также функцию «Подписать проект заявки ЭП», A_{223}, гарантирующей волеизъявление владельца учетной записи на передачу актива другим учетным записям и исключающей несанкционированный доступ (дополнительно к A_1) к активу в части распоряжения.

Электронная подпись представляет собой хэш-значение (хэш, *h*) данных *V-проекта заявки на транзакцию* (**Вход** A_{223}), зашифрованных закрытым ключом, ассоциированным с учетной записью владельца актива.

Подписание *V-проекта заявки на транзакцию ЭП*, A_{223}, в свою очередь, включает выполнение следующих функций (отдельная диаграмма не приводится).

1. «Хешировать» (создать хэш) данные *V-проекта заявки на транзакцию*, A_{2231}. Это гарантирует неизменность данных.
2. «Шифровать» хэш *V-проекта заявки на транзакцию* закрытым ключом владельца актива (создать ЭП), A_{2232}.
3. «Включить ЭП в проект» заявки на транзакцию, A_{2233}.

Выход функции A_{223} – *проект заявки на транзакцию с ЭП.*

Управление A_{233} – выход (санкция) функции проверки корректности проекта заявки на транзакцию A_{232} - *s & f-correctness.* **Механизм** A_{223}: ВС *i*-узла.

Завершает подготовку заявки на транзакцию функция «Передать проект заявки в БЧ-систему», A_{224}. При поступлении *Команды* владельца актива (**Управление** A_{224}) *Заявка на транзакцию* t_i^q поступает в БЧ-систему на почтовые ящики (п/я) всех *M*-узлов сети (мемпул неподтвержденных заявок *T*). В специальной литературе эта сущность иногда называется «неподтвержденная транзакция». Подтверждение транзакции происходит после включения заявки в БЧ (A_2, A_3).

С этого момента *заявка на транзакцию* t_i^q подлежит сложной многоуровневой и многофункциональной обработке системным процессом, предназначенным для включения неподтвержденных транзакций в реестр выполненных транзакций (в предметных терминах процесс **«Провести системную обработку волеизъявления»**, A_3). С технологической точки зрения этот процесс заключается в модификации БЧ, хранящего данные транзакций. Модификация состоит в создании нового *блока-кандидата* данных, его верификации и включении в существующую структуру данных (цепочку блоков). В БЧ-системе все узлы в любой момент времени находятся в одном из режимов: обработка данных заявок на транзакцию и создание нового блока-кандидата b_z^{cem} или верификация корректности нового блока-кандидата b_z^{cem}.

В функциональной модели это означает декомпозицию A_3 (диаграмма не приводится) и выполнение БЧ-системой, соответственно, процессов «**Создать блок-кандидат на включение в БЧ**», A_{31} и «**Верифицировать блок-кандидат**», A_{32}. Рассмотрим алгоритм и функционал БЧ-системы в этих режимах.

Мемпул заявок на транзакции поступает на **Вход** процесса A_{31}. **Выходом** A_{31} является распространенная *сетевая версия блока-кандидата* $\boldsymbol{b_z^{cem}}$ на включение в структуру БЧ.

Управление A_{31} выполняется по рассматриваемым ниже системным Правилам (алгоритму) создания блоков. **Механизм** A_{31}**:** в процессе сначала участвуют все *M*-узлы БЧ-системы (функции A_{311}-A_{313}), а затем *s*-узел ($s \ni M$), первым решивший сложную хэш-задачу, например, для заявки на транзакцию t_i^q (функции A_{314}-A_{315}).

Процесс A_{31} сложный, поскольку в его рамках решается не только собственно задача модификации структуры данных БЧ, но и задача поддержания целостности хронологии данных транзакций при обеспечении открытости БЧ-системы. Процесс A_{31} включает функции, представленные на рис. 6. Рассмотрим их.

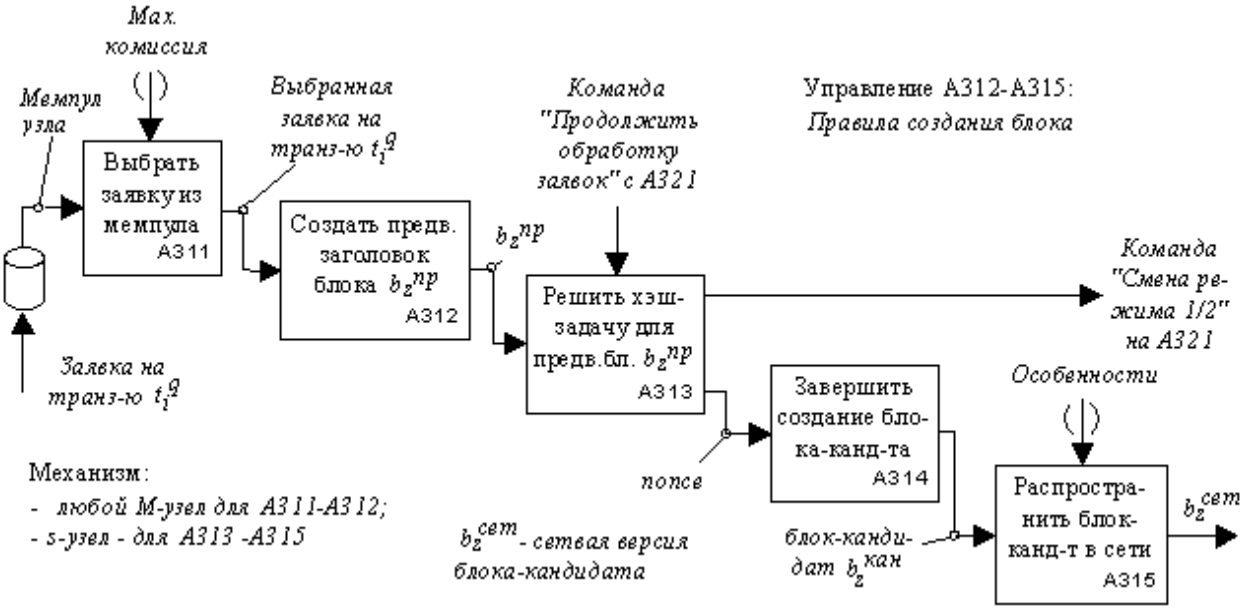

Рис. 6. Декомпозиция процесса «Создать блок-кандидат на включение в БЧ», A_{31}

Функция «Выбрать заявку из мемпула», A_{311} производит выбор заявки для дальнейшей обработки из множества заявок *T*, одномоментно находящихся в сети. Пусть это будет заявка на транзакцию t_i^q (Выход A_{311}). **Управление** A_{311} и **Механизм** A_{311} показаны на рис. 6. Сущность функции «Создать предв.<арительный> заголовок блока $b_z^{пр}$», A_{312} удобно рассматривать на ее декомпозиции (рис. 7).

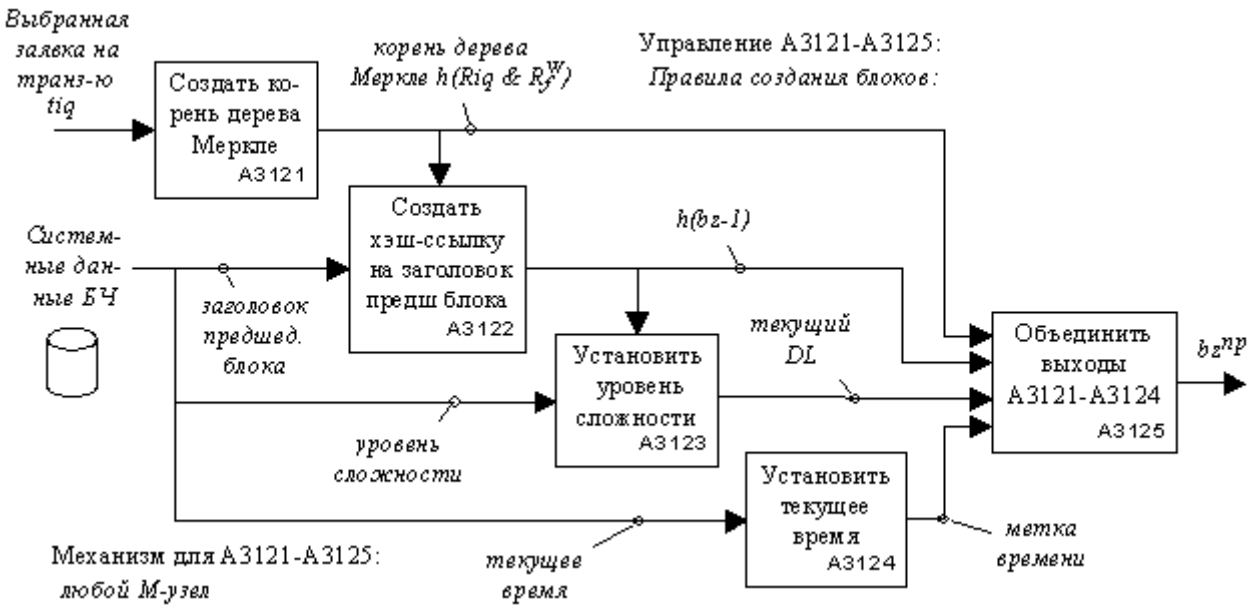

Рис. 7. Декомпозиция функции «Создать предв. заголовок блока $b_z^{пр}$», A 312

Функция «Создать корень дерева Меркле», A_{3121} заключается в хэшировании данных заявки и выполняется всеми *M*-узлами. **Выход**: хэш h (R_i^q & R_f^w), где R_i^q и R_f^w, соответственно, хэши данных заявок на транзакцию t_i^q и любой предшествующей ей, например, *w*-заявки из мемпула *T* от *f*-узла.

Функция «Создать хэш-ссылку на заголовок пред.<шествующего> блока», A_{3122}. **Выход**: хэш $h(b_{z-1})$. Блок $b_{z-1} < b_z \in B$, где B – упорядоченное множество блоков транзакций, БЧ.

Функции A_{3121}-A_{3122} создают основу структуры данных БЧ – упорядоченную цепочку заголовков блоков и деревьев Меркле, содержащих данные транзакций. После хэширования результатов A_{3121}-A_{3122} процесс добавления нового блока b_z в БЧ (A_{31}) мог бы быть закончен.

При этом архитектурные и технологические (хэширование) решения позволяют выявлять изменения: данных транзакции, ссылки на дерево Меркле, корня дерева Меркле, ссылки на заголовок блока, а также замену транзакции [2]. Эти изменения возможны без значительных затрат, так как предполагают лишь добавления хэш-ссылки, указывающей на заголовок нового блока в цепочке и объявление этой ссылки как новой головы цепочки. Но свойство открытости БЧ-системы (P2P-системы) актуализирует угрозу изменения хронологии данных транзакций. Поэтому необходимо обеспечить ее целостность при сохранении открытости системы. Для решения этой задачи использована идея понуждения интересанта к отказу от любых действий, вносящих изменения в БЧ. Это достигается целенаправленным созданием значительных организационно-технологических трудностей внесения изменений в структуру данных, преодоление которых делает изменения экономически нецелесообразными. Практическая реализация идеи обеспечивается выполнением следующих требований к БЧ-системе.

1. Выявление любых изменений хронологии данных транзакций (см. выше).

2. Создание усложненного алгоритма внесения изменений в хронологию транзакций, предполагающего редактирование и принудительную перезапись значительной части всей хронологии, начиная с места внесения изменения.

3. Решение сложных хэш-задач, различных для каждого заголовка блока.

Выполнение этих требований ведет к необходимости наличия значительных вычислительных мощностей для внесения изменений в хронологию транзакций. Конкретная реализация изложенных подходов к обеспечению целостности структуры данных БЧ осуществляется выполнением следующих функций.

1. «Установить требуемый уровень сложности», A_{3123}. Уровень сложности (difficulty level, DL) решения хэш-задачи, определяющий время для ее решения – системно устанавливаемый параметр, необходимый для последующего выполнения функции A_{313} (рис. 6).

2. «Установить текущее время» создания блока, A_{3124} (рис. 7).

3. «Решить хэш-задачу для <конкретного> предв.<арительного> <заголовка> блока $b_z^{пр}$», A_{313}, например, *s*-узлом (рис. 6).

Функция A_{313} имеет два **Выхода**: 1. Найденное решение хэш-задачи (nonce); 2. Команда *Смена режима 1/2*, поступающая в БЧ-систему. Эта команда останавливает функции A_{313} и актуализирует функции A_{321} (рис. 8) иных (не *s*)-узлов.

4. «Завершить создание блока-кандидата» $b_z^{кан}$, добавив одноразовый случайный код (nonce) к предварительному заголовку $b_z^{пр}$, A_{314}.

Реализация данного подхода включает в заголовок каждого блока БЧ следующие данные (информационная модель блока-кандидата):

- корень дерева Меркле, содержащего данные транзакции;
- хэш-ссылка на заголовок предыдущего блока;
- уровень сложности хэш-задачи;
- время начала решения хэш-задачи;
- одноразовый случайный код (nonce), решающий хэш-задачу.

Функция «Распространить блок-кандидат в сети», A_{315}. В БЧ (P2P)-системе каждый узел поддерживает собственную версию неизменяемого БЧ. Поэтому вновь созданный *s*-узлом блок-кандидат должен быть децентрализовано передан остальным узлам. **Вход**: $b_z^{кан}$, **Выход:** $b_z^{сет}$.

Управление A_{315}**.** Передача сообщений в P2P-системе имеет *Особенности*, затрудняющие их доставку адресатам. Эти трудности устраняются специальным алгоритмом и функционалом (в статье не рассматриваются).

Механизм A_{315}: транспортной основой БЧ-системы является Интернет.

Процесс «**Верифицировать блок-кандидат**», A_{32} необходим, поскольку в БЧ-систему возможно поступление некорректных блоков-кандидатов. **Входом** A_{32} является *сетевая версия блока-кандидата* $b_z^{сет}$. **Выходы** A_{32} (A_{321}, на рис. 8):

- *новый блок в БЧ* (модернизированный БЧ) с включенным новым блоком b_z в случае верификации блока-кандидата $b_z^{сет}$;
- команда блоку A_{313} (см. рис. 6) «*продолжить обработку заявок на транзакции*» (продолжить формирование новых блоков-кандидатов), остановленную *s*-узлом при нахождении *nonce* для транзакции t_i^p в противном случае.

Управление A_{32} рассматривается ниже. **Механизм:** процесс A_{32} выполняется всеми *M*-узлами БЧ-системы.

Декомпозиция A_{32} (рис. 8). Блок-кандидат $b_z^{сет}$ проходит процедуру «Проверить корректность блока-кандидата», A_{321}, которая определяется корректностью: данных транзакций (A_{3211}); заголовков блоков (A_{3212}); решения хэш-задачи (nonce), A_{3213}.

Управление. Правила проверки данных транзакций содержат условия формальной корректности, семантической (смысловой) корректности и авторизации (см. выше). Функция A_{321} актуализируется командой «Смена режима 1/2», поступающей с выхода функции A_{313} в момент нахождения *nonce s*-узлом.

Функция A_{321} имеет три выхода. **Выход 1.** В случае признания блока-кандидата $b_z^{сет}$ корректным, актуализируются функции БЧ-системы: «Добавить новый блок в свою копию БЧ», A_{322}; «Удалить заявку на транзакцию t_i^q», A_{323}; «Вознаградить *s*-узел», создавший новый блок, A_{324}.

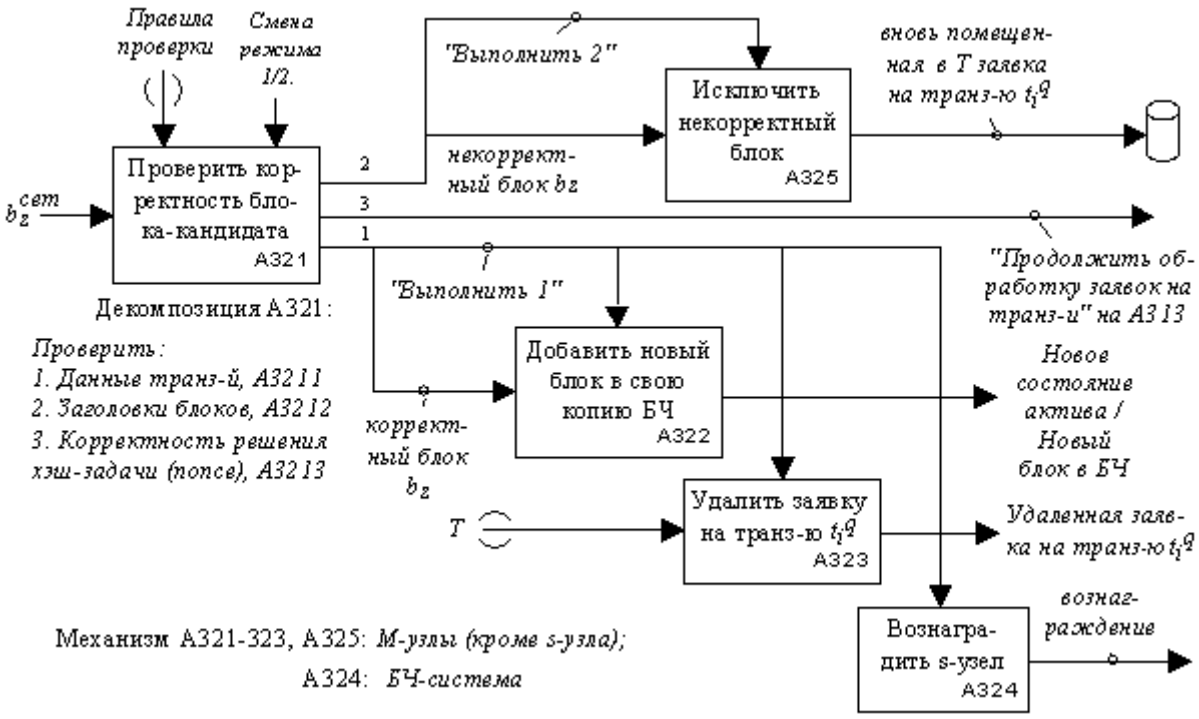

Рис. 8. Декомпозиция процесса «Верифицировать блок-кандидат», A_{32}

Функция A_{322} включает верифицированный блок b_z в копию БЧ. Функция A_{323} удаляет заявку на t_i^q-транзакцию из мемпулов всех узлов БЧ-системы (множества *T* заявок на транзакцию). Функция A_{324} предназначена для выявления, компенсации затрат (вознаграждение) и стимулирования (конкуренция) наиболее эффективных <владельцев> узлов, создающих и проверяющих блоки. Блок A_{324}, не имеет явного **Входа**, поскольку моделирует режим генерации БЧ-системой (**Механизм**) единицы *вознаграждения* (**Выход** *s*-узла).

Выход 2 A_{321}. В случае некорректности блока-кандидата $b_z^{сет}$ БЧ-система также по команде «*Продолжить обработку заявок на транзакции*» с **Выхода 3** возобновляет остановленный *s*-узлом режим формирования блоков-кандидатов (A_{313}), и узлы могут завершить этот этап добавлением своего блока. Данные транзакции t_i^q из неверифицированного блока возвращаются в мемпул для повторной обработки.

Актуализация (**Управление**) функций A_{322}-A_{325} производится признанием блока b_z корректным/некорректным (*команды* «*выполнить 1*» / «*выполнить 2*»). **Механизм** A_{321}-A_{324}, A_{325}: любой не *s*-узел.

Заключение

В заключение отметим, что в представленной модели отражены только основные функции БЧ-системы, необходимые для получения предметного результата. Состав этих функций согласуется с перечнем этапов проведения транзакций, представленным в специальной литературе: появление транзакции; опубликование транзакции в БЧ-сети; включение транзакции в новый блок; хэш нового блока засчитывается; новый блок предъявляется БЧ-сети; блок принимается сетью, выплачивается вознаграждение [8, 9].

За пределами детального рассмотрения (декомпозиции) остались «чисто» технологические функции, такие как «Передать проект заявки в БЧ-систему», A_{224}, «Вознаградить *xx*-узел», A_{324} и другие.

Также необходимо указать на отражение в модели ряда авторских представлений о логике работы БЧ-системы, в явном виде не описанной в литературе. Эти пробелы устранены введением, например, различных *команд*.

В целом, разработанную функциональную модель БЧ-технологии следует рассматривать как первую версию, требующую в соответствии с принципами IDEF0/SADT моделирования итеративной доработки.

Библиографический список

1. **Свон, М.** Схема новой экономики / М. Свон. – М.: Олимп-Бизнес, 2016. – 224 с.
2. **Дрешер, Д.** Основы блокчейна: вводный курс для начинающих в 25 небольших главах. / Д. Дрешер. – М.: ДМК Пресс, 2018. – 320 с.
3. **Марка, Д.А.** Методология структурного анализа и проектирования SADT (Structured Analysis & Design Technique) / Д. А. Марка, К. МакГоуэн. – М.: МетаТехнология, 1993. – 240 с.
4. Integration DEFinition for function modeling (IDEF0). Draft Federal Information Processing Standards Publication 183 ,1993 December 21. URL: http:idef.com/wp-content/uploads/2016/02/idef0.pdf.
5. РД IDEF0-2000. Методология функционального моделирования IDEF0. – М.: Госстандарт России, 2000. – 75 с.
6. Р 50.1.028-2001. Рекомендации по стандартизации. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. – М.: Госстандарт России, 2003. – 50 с.
7. **Лелу, Л.** Блокчейн от А до Я. Все о технологии десятилетия / Л. Лелу. – М.: Эксмо, 2018. – 256 с.
8. **Галкова, Е.В.** Биткойн – альтернатива инвестициям, криптовалюта или «стеклянные бусы» без реальной торговой стоимости? Обзор основных подходов регулирования / Е.В. Галкова // Законодательство. – 2016. – № 11. – С. 25-36.
9. **Максуров, А.А.** Майнинг как юридическая и информационная категория / А.А. Максуров // Актуальные проблемы экономики и права. – 2018. – Т. 12, № 2. – С. 256-265.

Дата поступления
в редакцию: 16.10.2018

V.Y. Karpychev

FUNCTIONAL MODELING (IDEF0) AS A METHOD OF RESEARCH ON BLOCKCHAIN TECHNOLOGY

Nizhny Novgorod state technical university n. a. R.E. Alekseev

Purpose: Development of blockchain technology model in tbe IDEF0 standard notation.
Methodology: The proposed blockchain technology model is based on Structured Analysis & Design Technique (SADT), a well-known methodology for modeling with objects of different nature. SADT/IDEF0 is applied to the known formal and verbal descriptions of the blockchain.
Value: The proposed model can be used in the development of original blockchain applications, as well as in foundational studies of the blockchain technology.
Research implications: The purpose of research is to further detail the model, represent structural and parametric characteristics of blockchain, identify model's possible vulnerabilities and limitations and propose ways to overcome them.

Keywords: asset management, functional modeling, blockchain technologies.