

Д. В. Жевнерчук, канд. техн. наук, доц., e-mail: zhevnerchuk@yandex.ru,
Нижегородский государственный технический университет им. Р. Е. Алексева

Обобщенный метод синтеза многокомпонентных интероперабельных структур на основе онтологии и недетерминированного конечного автомата

Обсуждается проблема синтеза многокомпонентных интероперабельных структур (МИС), которая относится к классу задач структурно-параметрического синтеза открытых информационных систем. Предложен обобщенный метод синтеза МИС, основанный на онтологическом представлении стандартизированных компонентов, их интерфейсов и формируемых многокомпонентных структур. Показано, что синтез МИС может быть сведен к задаче управления взаимодействующими процессами для организации распределенной обработки онтологии как разделяемого, распределенного ресурса. Предложена система событий, представленная в виде недетерминированного автомата, позволяющая формализовать и реализовать алгоритмы управления многопроцессной системой синтеза МИС.

Ключевые слова: открытая информационная система, компонент, интерфейс, многокомпонентная структура, интероперабельность, структурно-параметрический синтез, недетерминированный конечный автомат, распределенный алгоритм

Введение

В настоящее время все большее значение в области информационных технологий приобретает проблема автоматизации построения открытых информационных систем (ОИС), компоненты которых способны к взаимодействию, основанному на использовании информационно-коммуникационных технологий (ИКТ) [1]. Сам процесс построения ОИС сильно децентрализован и осуществляется независимыми группами технических специалистов, круг возможных специализаций которых достаточно широк, а именно: стандартизированные модели архитектурных решений; стандартизированные модели информационных процессов; стандартизированные интерфейсы и протоколы; стандартизированные модули (компоненты); интероперабельные, масштабируемые информационные системы.

Самым ярким примером открытой информационной системы является сеть Интернет, совмещающая большое число спецификаций, курируемых рабочими группами IETF, W3C и др. [1–3]. Разработчики решений для сети Ин-

тернет ориентируются на круг утвержденных нормативных документов и предлагают свои аппаратные/программные решения, поддерживающие стандартизированные интерфейсы и протоколы, а следовательно, обладающие необходимым уровнем интероперабельности для выполнения своих функций совместно с существующими компонентами сети. Поскольку для сети Интернет решения могут быть предложены любой командой, то имеет смысл говорить о группах разработчиков, географически распределенных по всей планете. Остается множество вопросов, связанных с инфокоммуникационной средой разработчиков, предоставляющей им необходимые минимальные сведения для эффективного участия в развитии сети Интернет, т. е. непосредственно касающихся автоматизации предпроектной, проектной деятельности.

При проектировании и разработке ОИС независимо от задач и областей применения все компоненты должны обладать стандартизированными интерфейсами и поддерживать стандартизированные протоколы взаимодействия. Информация о них размещается в справоч-

никах и справочных системах, которые, как правило, представляют собой варианты информационно-поисковых систем с модулями и интеграции в САПР или пакеты моделирования [4—6]. Такие системы эффективно используются в корпоративных разработках, не требующих аналитических обзоров, патентного поиска и поиска готовых встраиваемых решений и т. д.

Таким образом, можно констатировать факт отсутствия единых информационных пространств и коммуникационных сред, которые бы обеспечивали независимый ввод данных о компонентах и их свойствах, а также методах обеспечения их валидности в смысле соответствия известным спецификациям.

Другой стороной проблемы является отсутствие эффективных моделей и алгоритмов синтеза многокомпонентных структур на основе стандартизированных интерфейсов, поддерживающих информационные процессы.

Известна группа комбинаторно-логических методов структурного синтеза, которая ориентирована на решение задачи компонентной сборки технических систем. В основе этого подхода лежит хорошо организованный перебор в массиве решений, которые являются аналогами и прототипами.

Метод морфологического синтеза [7, 8], предложенный Ф. Цвикки, позволяет найти и систематизировать все возможные способы построения объекта, имеющие данное функциональное назначение. Для описания возможных комбинаций используется матрица инцидентности.

Метод синтеза с использованием N-дольных графов [9] позволяет группировать функции технического объекта по реализациям.

Структурный синтез по альтернативным деревьям [10, 11] содержит механизмы декомпозиции функций системы, при этом корень дерева соответствует основной технической функции класса объектов, висячие вершины (листья) представляют технические реализации, связи дерева описывают способы разбиения технических функций на подфункции.

Выделим следующие особенности комбинаторно-логических методов структурного синтеза: техническая система или процесс имеет структуру; техническая система принадлежит к некоторому классу объектов, имеющих одинаковое функциональное назначение; множество аналогов и прототипов обладает достаточной мощностью для того, чтобы поиск новых сочетаний в этом комбинаторном пространстве был результативен; составные части

объектов класса обладают "хорошими комбинаторными способностями".

Однако существующие методы структурного синтеза технических систем не могут быть напрямую применены для решения задачи синтеза открытых информационных систем, поскольку: описывают альтернативные фиксированные наборы компонентов; отсутствует формализация интероперабельности и иных ограничений на компоненты; требуют модификации для использования в прикладных решениях, ориентированных на единые информационные пространства.

Синтез ОИС иерархичен и адаптивен и направлен на формирование многокомпонентных структур, элементы которых представляют собой блоки (черные ящики), обладающие стандартизированными свойствами (параметрами), для каждого из которых определяется свой домен (базовый тип плюс ограничения, задаваемые диапазоном, перечислением или шаблоном). Параметры блоков, связанные ассоциативно с точками доступа (вход, выход), определяют интерфейсы компонентов ОИС. На системном уровне множество блоков не обладает структурой (является неструктурированным). Процесс структурирования заключается в формировании цепочек компонентов, причем два любых соседних компонента могут быть сопряжены непосредственно по входам/выходам.

Получаемые многокомпонентные структуры могут быть использованы при онтологическом описании информационных процессов, требований/ограничений, технических (программных/аппаратных) систем. Известно, что базовым механизмом синтеза элементов онтологии является логический вывод, базирующийся на теоретико-множественных операциях и дескрипционной логике, выполняемый на графах. Таким образом, при построении алгоритмов синтеза ОИС неизбежно возникают проблемы выявления классов NP-полных задач.

Еще одной особенностью является то, что сам граф является динамическим, т. е. в общем случае для двух любых компонентов время их регистрации в системе отличается. Можно говорить о потоке заявок на регистрацию блоков и их свойств, носящем стохастический характер, причем регистрация каждого нового компонента приводит к необходимости реструктурирования графа, формированию новых узлов и новых связей. *В этом случае алгоритмическая обработка осуществляется системой обработчиков, для каждого из которых граф представляет собой разделяемый ресурс.*

Регистрация каждого компонента приводит к инициализации цепочки преобразований графовой структуры, т. е. к изменению его состояния.

Таким образом, ввиду стохастической природы потока заявок на регистрацию компонент, конечного числа преобразователей графа, а также с учетом того факта, что хранение графовой структуры носит распределенный характер, синтезирующий алгоритм может быть представлен посредством системы обработчиков (процессов), взаимодействие (синхронизация, контроль критических областей, связанных с разделяемым ресурсом, обработка прослушивания, прием и обработка заявок на регистрацию компонент) может быть представлено недетерминированным конечным автоматом.

1. Постановка задачи. Требования и ограничения синтезируемой системы

На основе вышеизложенного может быть сформулирована постановка задачи построения информационных моделей, обладающих свойствами расширяемости и масштабируемости, обеспечивающих независимый ввод, редактирование и хранение данных о компонентах ОИС, их свойствах и интерфейсах, регламентируемых известными открытыми спецификациями, а также алгоритмов синтеза многокомпонентных структур на их основе.

Расширяемость информационной модели заключается в способности добавления новых компонент и многокомпонентных структур без необходимости внесения изменений в существующие элементы и связи между ними. В первую очередь это достигается за счет слоя метауровня, на котором должны быть определены все базовые классы и связи между ними, а также за счет выделения отдельных информационных уровней, позволяющих локализовать синтез системных свойств на основе обработки сильносвязанных элементов и разграничить обработку слабосвязанных элементов. Кроме того, необходимо выполнить отдельным уровнем формализацию спецификаций, благодаря которому свойства и интерфейсы каждого нового компонента будут отображаться на пространство эталонных (стандартизированных) свойств и интерфейсов. Это позволит реализовать пополнение модели из множества независимых источников.

Масштабируемость информационной модели заключается в способности сформиро-

вать результаты синтеза многокомпонентных структур за интервал времени, в течение которого они не теряют своей актуальности. В условиях стохастического потока заявок на регистрацию компонент, а также следующего за этим синтеза интерфейсов и многокомпонентных структур масштабируемость обеспечивается фрагментацией информационной модели с последующей ее параллельной обработкой.

2. Онтологическое моделирование многокомпонентных интероперабельных структур

В работе [12] был предложен онтологический каркас ОИС, регламентирующий множество базовых концептов и ролей, описывающих взаимосвязи объектов ОИС на уровне нормативной документации, информационных процессов (бизнес-процессов), программно-аппаратных комплексов. В каркасе введены такие концепты, как *Interface* (Интерфейс), *Parameter* (Параметр), *Restriction* (Ограничение), причем ограничения могут быть двух типов: перечисления и диапазон, а также определены роли, связывающие блоки документов, определяющих интерфейсы с блоками стандартизированных профилей ОИС. Введенный концепт *Interface Profile* (Профиль интерфейса) предназначен для группирования параметров, определенных стандартизированным профилем ОИС. В такой постановке возникает избыточность, связанная с тем, что требуется хранить данные о блоках спецификаций и блоках стандартизированных профилей, которые являются одновременно и блоками спецификаций. Дополнительно стоит отметить тот факт, что ряд ограничений определяется функциями, которые формируют подмножества базового типа, либо исключаемые из области определения параметра, либо принимаемые за область определения.

Для устранения этого недостатка был введен концепт *Domain*, индивиды которого формируются, с одной стороны, на основе параметров, определенных в блоках спецификаций, а с другой — они связаны с индивидами свойств программных или аппаратных компонент. На рис. 1 показан фрагмент семантической модели, отражающий связь доменов, определенных в спецификациях, с областями определения свойств компонент и интерфейсами.

Свойства компонент, для которых такая связь существует, назовем *стандартизированными*. Если все свойства компонента стандар-

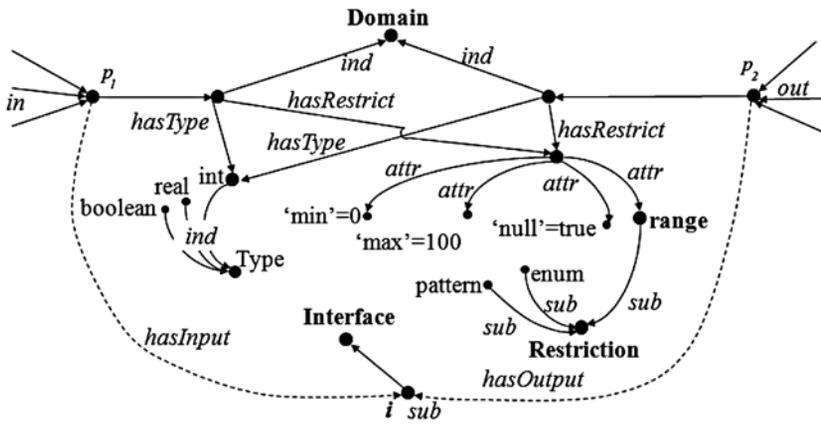


Рис. 1. Связь доменов с областями определения свойств и интерфейсами

тизируются, то компонент является стандартизированным.

Итак, каждый компонент, добавляемый в систему, дополняется свойствами, которые привязываются к точкам доступа типа вход/выход и стандартизируются посредством системы концептов и ролей, определенных в онтологическом каркасе, о котором говорилось ранее. Для каждого параметра блока синтезируется индивид-Домен, связанный с индивидами, определяющими базовый тип, и возможные ограничения, такие как диапазон (*range*), перечисление (*enum*), шаблон (*pattern*). На рис. 1 определены два параметра: p_1 привязан к точке доступа типа "вход" одного блока, а p_2 привязан к точке доступа типа "выход" другого блока. Считаем, что существует функцио-

нальное отображение $f: f(p_1) = s_1, f(p_2) = s_2$, где s_1, s_2 — ограничения параметров, определяемые спецификациями. Поскольку для s_1, s_2 определены базовый тип и ограничения, то они определены также и для p_1, p_2 . Пусть d_1, d_2 — области допустимых значений p_1, p_2 , задаваемые базовым типом данных x и ограничениями r , вводимыми на x . Тогда, если для p_1, p_2 справедливо $f(p_1) = f(p_2)$, то d_1, d_2 обладают общим базовым типом и общими ограничениями на нем.

Так как p_1 связан с точкой доступа типа "вход", а p_2 с точкой доступа типа "выход", то возможен информационный процесс между связанными с ними блоками и можно выполнить переход от доменов к единому интерфейсу i . Переход осуществляется забывающим функтором, скрывающим данные о доменах [13].

На рис. 2 представлена схема *ABox* [14], иллюстрирующая систему концептов и ролей, используемых при регистрации блоков, их свойств и доменов, а также их стандартизации. Для простоты рассматривается множество, состоящее из четырех блоков b_1, \dots, b_4 , каждый из которых обладает двумя свойствами с определенными доменами.

Свойства привязываются к точкам доступа блоков посредством ролей "hasProp" (свойство принадлежит блоку), "in", "out" (свойство привязано к точке доступа типа "вход" или "выход"). Стандартизация свойств осуществляется путем их функционального отображения на параметры спецификаций, вследствие чего с каждым свойством блока сопоставляется копия домена, определенного в спецификации и связанного с результирующим параметром той же спецификации. В результате стандартизации между свойством и его базовым блоком появляется свойство "hasSTProp" (свойство было сопоставлено с параметром спецификации, и поэтому оно стандартизировано).

На следующем этапе происходит сопоставление доменов интерфейсных прошедших стандартизацию свойств зарегистрированных блоков, в результате чего формируется слой интерфейсов [13] (рис. 3).

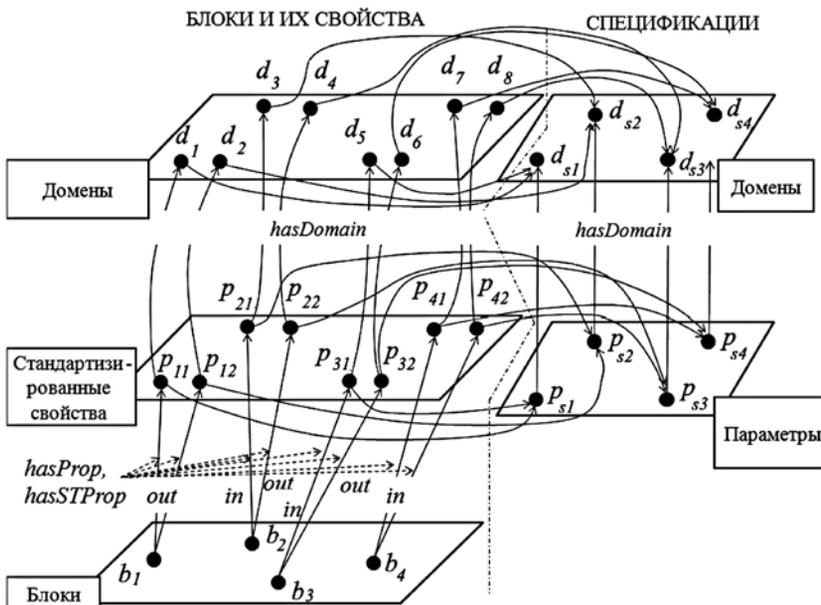


Рис. 2. Регистрация и стандартизация блоков и их свойств

На текущем уровне выполняется переход от блоков с доменными ограничениями по входу и выходу к компонентам и их интерфейсам, причем домен, привязанный к точке входа/выхода, отображается на интерфейс лишь в том случае, если через связанную с ним точку доступа существует канал обмена данными между двумя и более блоками. Далее выполняется синтез связей между компонентами *compatible* (существование канала передачи данных от компонента источника к компоненту приемнику).

Синтез многокомпонентных структур осуществляется посредством фильтрации построенной структуры *compatible*-связности компонентов.

Итак, процесс синтеза многокомпонентных структур является иерархическим, при переходе между соседними уровнями формируются новые обобщающие конструкции:

- при переходе от доменов к интерфейсам (обработка терминологической аксиомы машиной логического вывода);
- при переходе от интерфейсов к совместимости (переход к транзитивному замыканию ролей *in*, *out* на *compatible*);
- при переходе от сети совместимых компонентов к *n*-компонентным структурам.

Как было отмечено выше, структуры совместимых компонент используются на уровнях информационных процессов, требований/ограничений, техническом (программном/аппаратном) и могут образовывать межуровневые структуры совместимых компонент.

Выделим следующие межуровневые связи:

1. "Информационный процесс" — "Технический компонент" (технический компонент автоматизирует информационный процесс, обладает подмножеством интерфейсов компонента информационного процесса).

2. "Требование/Ограничение" — "Технический компонент" (технический компонент удовлетворяет требованиям/ограничениям, обладает подмножеством интерфейсов компонента требований/ограничений).

3. "Требование/Ограничение" — "Информационный процесс" (ин-

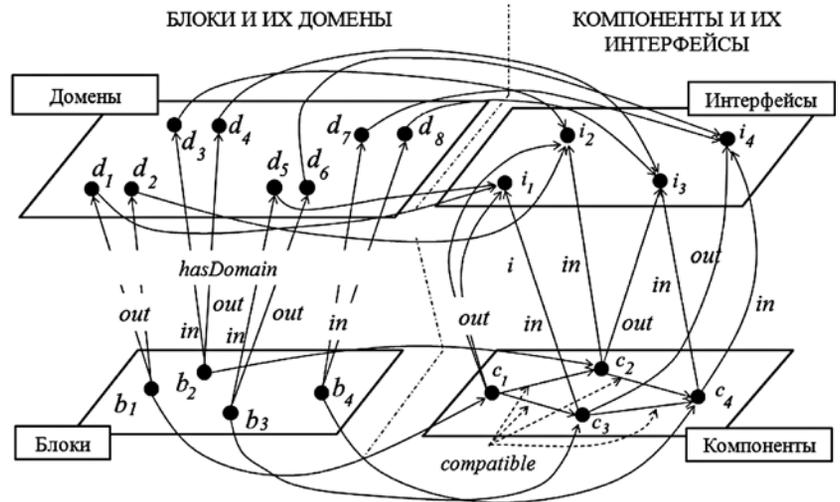


Рис. 3. Синтез интерфейсов и компонентов

формационный процесс удовлетворяет требованиям/ограничениям, обладает подмножеством интерфейсов компонента требований/ограничений).

Вместе с тем синтез целевой многокомпонентной интероперабельной структуры (МИС) представляет собой иерархический процесс, каждый уровень которого обладает независимой системой событий, формализуемой недетерминированным конечным автоматом (НДА) (рис. 4).

Операции сопряжения компонентов на одном уровне и между уровнями имеют следующие ключевые отличия. Если для компонента одного из уровней не существует ни одного со-

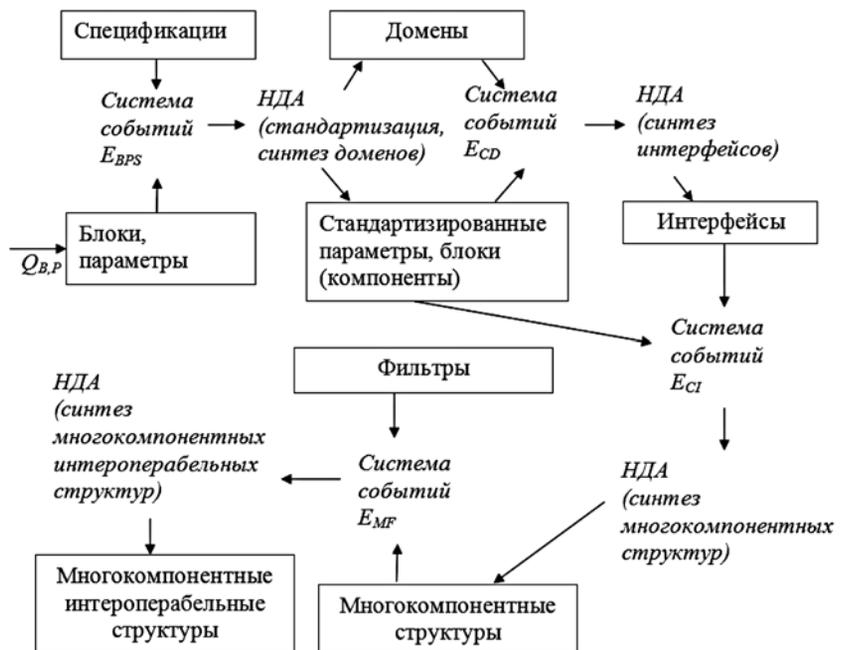


Рис. 4. Общая схема синтеза МИС на основе НДА

вместимого компонента другого уровня, то этот компонент не является элементом результирующей структуры и удаляется из нее. Один уровень является фильтрующим, а другой — фильтруемым, а операция межуровневого совмещения компонентов называется фильтрацией.

Фильтрующими являются:

- "информационный процесс" в связи "информационный процесс \leq > технический компонент";
- "требование/ограничение" в связи "требование/ограничение \leq > технический компонент";
- "требование/ограничение" в связи "требование/ограничение \leq > информационный процесс".

3. Формализация и построение алгоритмов управления взаимодействующими процессами синтеза МИС на основе НДА

Синтез МИС включает два основных этапа: сопряжение компонентов в пределах одного уровня (синтез многокомпонентных структур), фильтрация или сопряжение компонентов, принадлежащих различным уровням (синтез многокомпонентных интероперабельных структур).

Первый этап разбивается на следующие шаги: а) синтез интерфейсов; б) синтез пар блоков, сопряженных через интерфейсы; в) синтез комплексных блоков (блоки, определяемые подпоследовательностями сопряженных блоков, причем у комплексных блоков входной интерфейс совпадает с входным интерфейсом первого блока последовательности, а выходной интерфейс — с выходным интерфейсом последнего блока подпоследовательности сопряженных блоков).

Второй этап представлен следующими шагами: а) синтез фильтрующих интерфейсов; б) фильтрация в пределах межуровневых связей.

Синтез целевой ОИС является результатом композиции интероперабельных структур.

Базовой моделью, отражающей структурные и параметрические аспекты МИС, является граф, и реализация выделенных шагов сводится к решению задач: изменения структуры подграфов (добавление, изменение, удаление ребер и узлов); поиск подграфов; кластеризация на одном графе; кластеризация на множестве графов.

Таким образом, алгоритмическое обеспечение синтеза МИС представляет собой комплекс информационно-поисковых и CUD-ал-

горитмов (CREATE, UPDATE, DELETE) на распределенных графовых структурах большой размерности.

В работах А. А. Шальто [15—17], Н. П. Вашкевича [18] заложены теоретические основы параллельных алгоритмов обработки данных большой размерности с позиций конечных недетерминированных автоматов, и тогда переход от онтологической модели к событийной позволяет: а) выполнять управляемую событиями инициализацию алгоритмов (необходимо из-за существования стохастических потоков заявок на регистрацию новых компонентов, а также потоков запросов на формирование целевых ОИС); б) сгруппировать алгоритмы по переходам между соседними уровнями; в) сгруппировать алгоритмы по вычислительным процессам; г) организовать управление взаимодействующими процессами синтеза МИС в распределенной вычислительной среде.

Рассмотрим подробнее НДА регистрации, стандартизации блоков и их свойств и синтеза доменов. В начальный момент времени автомат (рис. 5) обладает активными событиями ожидания поступления данных о компонентах. Индекс i в обозначении состояния является идентификатором ожидающего процесса. Если в этом состоянии автомат на входе получает вектор данных о компоненте (о свойствах блоков), то он переходит к нескольким событиям:

1. Событие Reg связано с синтезом индивидов "Свойство", а также с формированием роли, соединяющей точки доступа блока со свойствами. Если свойство составное, то формируются роли $hasComposProp$ и $hasSubProp$.

2. Событие W_{p_i} связано с активизацией следующего потока в очереди потоков (если существуют свободные потоки).

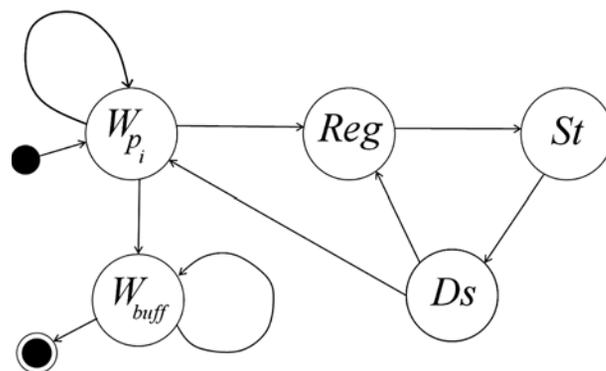


Рис. 5. НДА регистрации и стандартизации блоков и их свойств

3. Событие W_{buff} связано с буферизацией входящих запросов (если все потоки выполнят алгоритмическую обработку онтологии, нет ожидающих).

По завершению регистрации свойств компонентов i -м потоком автомат переходит к событию St , связанному с поиском блока спецификации s_p . Также событие может быть инициировано в результате пользовательского ввода данных о части спецификации, содержащей описание необходимых параметров, ограничений и правил интерпретации. На выходе также передается индекс потока, завершившего регистрацию свойств.

Далее инициируется событие DS синтеза доменов стандартизированных параметров, из которого автомат может вернуться к событию Reg , если на входе наблюдается значение буфера заявок > 0 , или к событию W_{p_i} , если буфер пуст. Также, в случае пустого буфера, автомат сбрасывает событие W_{buff} .

Другие НДА используются на этапах синтеза интерфейсов и многокомпонентных структур. Их подробное рассмотрение остается за рамками статьи.

Заключение

В работе предложен обобщенный метод синтеза МИС, основанный на онтологии компонентов, их интерфейсов, а также на формализации и реализации алгоритмов управления взаимодействующими процессами, осуществляющими операции синтеза на всех уровнях онтологического представления.

Предложена многоуровневая онтологическая модель МИС, обеспечивающая переход к системе событий, что позволяет локализовать алгоритмическую обработку базового графа, а также:

а) выполнять управляемую событиями инициализацию алгоритмов (необходимо из-за существования стохастических потоков заявок на регистрацию новых компонентов, а также потоков запросов на формирование целевых ОИС);

б) сгруппировать алгоритмы по переходам между соседними уровнями;

в) сгруппировать алгоритмы по вычислительным процессам;

г) организовать управление взаимодействующими процессами синтеза МИС в распределенной вычислительной среде.

1. Гуляев Ю. В., Журавлев Е. Е., Олейников А. Я. Методология стандартизации для обеспечения interoperability информационных систем широкого класса. Аналитический обзор // Журнал радиоэлектроники. 2012. № 3. URL: jre.cplire.ru/jre/Mar/12/2/text/pdf (дата обращения: 14.06.2018).
2. RFCs. Memos in the RFC document series contain technical and organizational notes about the Internet. URL: <http://www.ietf.org/standards/rfcs/> (дата обращения: 14.06.2018).
3. W3C Strategic Highlights. Technical Report/May. 2018. URL: <https://www.w3.org/2018/05/w3c-highlights/> (дата обращения: 14.06.2018).
4. ГОСТ Р ИСО/МЭК 9594-1—98 Информационная технология (ИТ). Взаимосвязь открытых систем. Справочник. Часть 1. Общее описание принципов, моделей и услуг. URL: <http://docs.cntd.ru/document/1200028703/> (дата обращения: 14.06.2018).
5. Зиченко Ю. В., Голобородько А. А. Обзор современных систем автоматизированного проектирования // Потенциал современной науки. 2016. № 4 (21). С 68—71.
6. Системы имитационного моделирования: выбираем подходящую. ГК Ланит, 2018. URL: <https://habr.com/company/lanit/blog/351870/> (дата обращения: 14.06.2018).
7. Одрин В. М., Картавов С. С. Морфологический анализ систем. Киев: Нукова Думка, 1977. 148 с.
8. Закревский А. Д. Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 511 с.
9. Божко А. Н., Толпаров А. Ч. Структурный синтез на элементах с ограниченной сочетаемостью // Наука и Образование: электронное научно-техническое издание. 2004. № 5.
10. Сафронов В. В. Методы оптимизации структур сложных систем. Саратов: СВВКИУ РВ, 1993. 94 с.
11. Сафронов В. В. Основы системного анализа: методы многовекторной оптимизации и многовекторного ранжирования. Саратов: Научная книга, 2009. 329 с.
12. Жевнерчук Д. В. Онтологический каркас поддержки профилирования вычислительных систем // Системы управления и информационные технологии. 2014. № 2.1 (56). Москва—Воронеж: ИПУ РАН — ВГТУ. С. 187—190.
13. Ломакина Л. С., Жевнерчук Д. В. Синтез открытых информационных систем с использованием алгебраических структур как моделей // Фундаментальные исследования. 2017. № 10. С. 29—33.
14. Бездушный А. А. Математическая модель системы интеграции данных на основе онтологий // Вестник НГУ. Сер. "Информационные технологии". Новосибирск. 2008. Т. 6, Вып. 2. С. 15—40.
15. Лукин М. А., Шалыто А. А. Разработка и автоматическая верификация параллельных автоматных программ // Информационно-управляющие системы. 2013. № 5 (66). С. 43—50.
16. Pavlov A., Sokolov B., Pashchenko A., Shalyto A., Maklakov G. Models and methods for multicriteria situational flexible reassignment of control functions in man-machine systems // Proceedings of the 8th IEEE International Conference Intelligent Systems (IEEE IS'2016). September 4—6 2016, Sofia, Bulgaria.
17. Dubinin V., Vyatkin V., Shalyto A. Formal modeling and verification of IEC 61499 function blocks on the basis of transition systems // Proceedings of International Siberian conference on control and communications (SIBCON 2016). M., 2016. 4 p. DOI: 10.1109/SIBCON.2016.7491701.
18. Вашкевич Н. П., Бикташев Р. А. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации. Пенза: Изд-во ПГУ, 2016. 394 с.

A Generalized Method for Synthesizing Multicomponent Interoperable Structures Based on Ontology and a Nondeterministic Finite State Machine

The work is devoted to the problem of synthesis of multicomponent interoperable structures (MIS), which belongs to the class of open information systems structurally-parametric synthesis problems. A generalized method for synthesizing MIS based on the standardized components ontological representation, their interfaces and formed multicomponent structures is proposed. It is shown that the synthesis of MIS can be reduced to the task of managing interacting processes for the organization of distributed processing of ontology as a shared, distributed resource. A system of events is proposed, presented in the form of a nondeterministic finite state machine, which allows one to formalize and implement algorithms for controlling the multiprocess synthesis system for MIS.

Keywords: open information system, component, interface, multicomponent structure, interoperability, structural-parametric synthesis, nondeterministic finite state machine, distributed algorithm

DOI: 10.17587/it.25.67-74

References

1. Gulyaev Y. V., Zhurtavlev E. E., Oleinikov A. Y. *Metodologiya standartizatsii dlya obespecheniya interoperabel'nosti informacionnih sistem shirokogo klassa. Analiticheskii obzor* (Methodology of standardization for providing interoperability of information systems of a wide class. Analytical review), *Zhurnal Radioelektroniki*, 2012, no. 3, available at: jre.cplire.ru/jre/Mar/12/2/text/pdf. (date of access: 14.06.2018) (in Russian).
2. RFCs. Memos in the RFC document series contain technical and organizational notes about the Internet, available at: <http://www.ietf.org/standards/rfcs/> (date of access: 14.06.2018).
3. W3C Strategic Highlights. Technical Report/May, available at: <https://www.w3.org/2018/05/w3c-highlights/> (date of access: 14.06.2018).
4. GOST R ISO / MEK 9594-1—98. *Informacionnaya tehnologiya (IT). Vzaimosvyaz' otkritih sistem. Spravochnik. Chast' 1. Obschee opisanie principov, modelei I uslug* (GOST R ISO / IEC 9594-1—98 Information technology (IT). Interconnection of open systems. Directory. Part 1. General description of principles, models and services), available at: <http://docs.cntd.ru/document/1200028703/> (date of access: 14.06.2018) (in Russian).
5. Zinchenko Y. V., Goloborodko A. A. *Obzor sovremennih sistem avtomatizirovannogo proektirovaniya* (Overview of modern CAD systems), *Potencial Sovremennoi Nauki*, 2016, no. 4 (21), pp. 68—71 (in Russian).
6. *Sistemi imitacionnogo modelirovaniya: vibiraem podhodnyashchuyu* (Simulation systems: choose the appropriate), GK Lanit, 2018, available at: <https://habr.com/company/lanit/blog/351870/> (date of access: 14.06.2018) (in Russian).
7. Odrin V. M., Kartavov S. S. *Morfologicheskii analiz sistem* (Morphological analysis of systems), Kiev, Nukova Dumka, 1977, 148 p. (in Russian).
8. Zakrevskii A. D. *Algoritmi sinteza diskretnih avtomatov* (Algorithms for the synthesis of discrete automata), Moscow, Nauka, 1971, 511 p. (in Russian).
9. Bozhko A. N., Tolparov A. Ch. *Strukturnii sintez na elementah s ogranichennoi sochetaemost'yu* (Structural synthesis on elements with limited compatibility), *Nauka I Obrazovanie: elektronnoe nauchno-tehnicheskoe izdanie*, 2004, no. 5 (in Russian).
10. Saphronov V. V. *Metodi optimizatsii struktur slozhnih sistem* (Methods for optimizing the structures of complex systems), Saratov, SVVKIU RV, 1993, 94 p. (in Russian).
11. Saphronov V. V. *Osnovy sistemnogo analiza: metody mnogo-vektornoj optimizatsii i mnogovektornogo ranzhirovaniya* (Fundamentals of system analysis: methods of multi-vector optimization and multi-vector ranking: monograph), Saratov, Nauchnaya kniga, 2009, 329 p. (in Russian).
12. Zhevnerchuk D. V. *Ontologicheskii karkas podderzhki profilirovaniya vichislitel'nykh sistem* (Ontological framework for supporting the profiling of computing systems), *Sistemi Upravleniya i Informacionnie Tehnologii*, 2014, no. 2.1 (56), Moscow — Voronezh, IPU RAN — VGTU, pp. 187—190 (in Russian).
13. Lomakina L. S., Zhevnerchuk D. V. *Sintez otkritih informacionnih sistem s ispol'zovaniem algebraicheskikh struktur kak modelei* (Synthesis of open information systems using algebraic structures as models), *Fundamental'nie Issledovaniya*, 2017, no. 10, pp. 29—33 (in Russian).
14. Bezduzhnii A. A. *Matematicheskaya model sistemi integratsii danih na osnove ontologii* (Mathematical model of data integration system based on ontologies), *Vestnik NGU, ser. "Informacionnie tehnologii"*, Novosibirsk, 2008, vol. 6, no. 2, pp. 15—40 (in Russian).
15. Lukin M. A., Shalyto A. A. *Razrabotka i avtomaticheskaya verifikatsiya parallel'nykh avtomatnykh program* (Development and automatic verification of parallel automata programs), *Informacionno-Upravlyayushchie Sistemy*, 2013, no. 5 (66), pp. 43—50 (in Russian).
16. Pavlov A., Sokolov B., Pashchenko A., Shalyto A., Maklakov G. Models and methods for multicriteria situational flexible reassignment of control functions in man-machine systems, *Proceedings of the 8th IEEE International Conference Intelligent Systems (IEEE IS'2016)*, September 4—6 2016, Sofia, Bulgaria.
17. Dubinin V., Vyatkin V., Shalyto A. Formal modeling and verification of IEC 61499 function blocks on the basis of transition systems, *Proceedings of International Siberian conference on control and communications (SIBCON 2016)*, Moscow, 2016, 4 p. DOI: 10.1109/SIBCON.2016.7491701.
18. Vashkevich N. P., Biktashev R. A. *Nedeterminirovannye avtomaty I ikh ispol'zovanie dlya realizatsii parallel'noi obrabotki informatsii* (Non-deterministic automata and their use for the implementation of parallel processing systems: Monograph), Penza, Publishing house of PGU, 2016, 394 p. (in Russian).