

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 004.82: 004.4'2

В. Н. Дубинин, В. В. Вяткин

СЕМАНТИЧЕСКИЙ АНАЛИЗ ОПИСАНИЙ СИСТЕМ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМИ ПРОЦЕССАМИ НА ОСНОВЕ СТАНДАРТА IEC 61499 С ИСПОЛЬЗОВАНИЕМ ОНТОЛОГИЙ¹

Аннотация. Представлена Web-онтология систем управления промышленными процессами, построенных в соответствии с новым международным стандартом IEC 61499. Математической основой онтологии являются дескриптивная логика и логика хорновских дизъюнктов, а языками реализации – OWL DL и SWRL. Предлагается метод семантического анализа описаний систем управления на основе определения в онтологии так называемых «семантически правильных» классов.

Ключевые слова: стандарт IEC 61499, функциональный блок, онтология, семантический анализ, дескриптивная логика, SWRL.

Abstract. In the paper a Web-ontology of industrial process control systems based on the new international standard IEC 61499, is presented. The mathematical footing of the ontology is a combination of description logic and Horn logic, while the implementation languages are OWL DL, and SWRL, correspondingly. An approach to semantic analysis of control systems descriptions on the basis of defining in the ontology so-called “semantically right” classes is proposed. Illustrative examples are given.

Keywords: standard IEC 61499, function block, ontology, semantic analysis, description logic, SWRL.

Введение

В настоящее время в связи с появлением и внедрением в практику мультиагентных систем и развитием концепции семантического Web большое значение приобретает онтологическое представление знаний [1]. Под *онтологией* понимается формальное представление множества концептов внутри домена и отношений между этими концептами [1]. В одном из сценариев, пока еще мало исследованном, онтологии используются при проведении *семантического анализа*. В дальнейшем будем касаться только той области семантического анализа, которая относится к языкам, включая языки программирования. Следует также отметить, что семантический анализ является одной из фаз работы (традиционного) компилятора.

¹ Работа выполнена в рамках аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2009–2010 годы)», № гос. регистрации НИР 01200952061.

Международный стандарт IEC 61499 [2] определяет путь для создания систем управления промышленными процессами нового поколения. Это интеллектуальные распределенные компонентно-базированные системы на основе функциональных блоков (ФБ). Функциональный блок (ФБл) является основным артефактом проектирования в стандарте IEC 61499. В данной работе рассматривается онтологический подход к описанию систем управления на основе стандарта IEC 61499 с использованием дескриптивной логики (ДЛ) [3] и логики хорновских дизъюнктов [4]. База знаний системы представления знаний (СПЗ) на основе ДЛ состоит из двух компонентов: TBox и ABox. Компонент TBox вводит терминологию, иными словами, словарь предметной области, в то время как компонент ABox содержит утверждения об именованных представителях (экземплярах) концептов в терминах словаря. Составными частями TBox являются концепты и роли.

На основе ДЛ был разработан язык Web-онтологий OWL [5], являющийся одной из основных составных частей семантического Web. Для увеличения выразительной силы в совокупности с ДЛ используется логика хорновских дизъюнктов, являющаяся фрагментом логики первого порядка [4]. Хорновские дизъюнкты могут представляться в виде правил языков логического программирования, например, Prolog [4, 6] или Datalog. В качестве языка правил в семантическом Web используется язык SWRL [7]. Для того чтобы формализм OWL DL + SWRL был разрешим, следует использовать так называемые «ДЛ-безопасные» правила (DL-safe rule [8]).

В данной работе (частично) представляется онтология систем управления промышленными процессами, построенных в соответствии с новым международным стандартом IEC 61499, и предлагается метод семантического анализа описаний систем, основанный на определении в онтологии так называемых «семантически правильных» классов. Подходы, предложенные в статье, будут демонстрироваться на примере системы ФБ (из работы [6]), приведенной на рис. 1.

2. Описание объектного каркаса онтологии

На рис. 2–5 представлен объектный каркас онтологии систем управления на основе IEC 61499 в виде интенциональных семантических сетей [9]. На этих диаграммах концепты изображаются в виде овалов, а роли – в виде стрелок, соединяющих овалы. В дальнейшем концепты также будем называть классами, а роли – отношениями или свойствами.

Описание типов функциональных блоков

В соответствии со стандартом IEC 61499 различаются три основных типа функциональных блоков (FBType): базисный, составной и сервисный интерфейсный (СИФБ). Данный факт представлен на рис. 2. Следует различать ФБл-тип и ФБл-экземпляр, именуемый в стандарте компонентным ФБл.

Описание интерфейсов

Каждый ФБ имеет интерфейс, представляющий собой набор событийных и информационных входов и выходов. Несмотря на определенную схожесть, будем различать интерфейсы ФБ-типов и компонентных ФБ. Пример интерфейса ФБл-типа приведен на рис. 1 (слева). На рис. 3 представлена семантическая сеть для определения интерфейсов ФБ-типов. Из рис. 3 видно,

что в состав интерфейса ФБл-типа входят событийные входы и выходы, входные и выходные переменные, а также сокететы (SocketsOfFBType) и штекеры (PlugOfFBType), относящиеся к адаптерам [2].

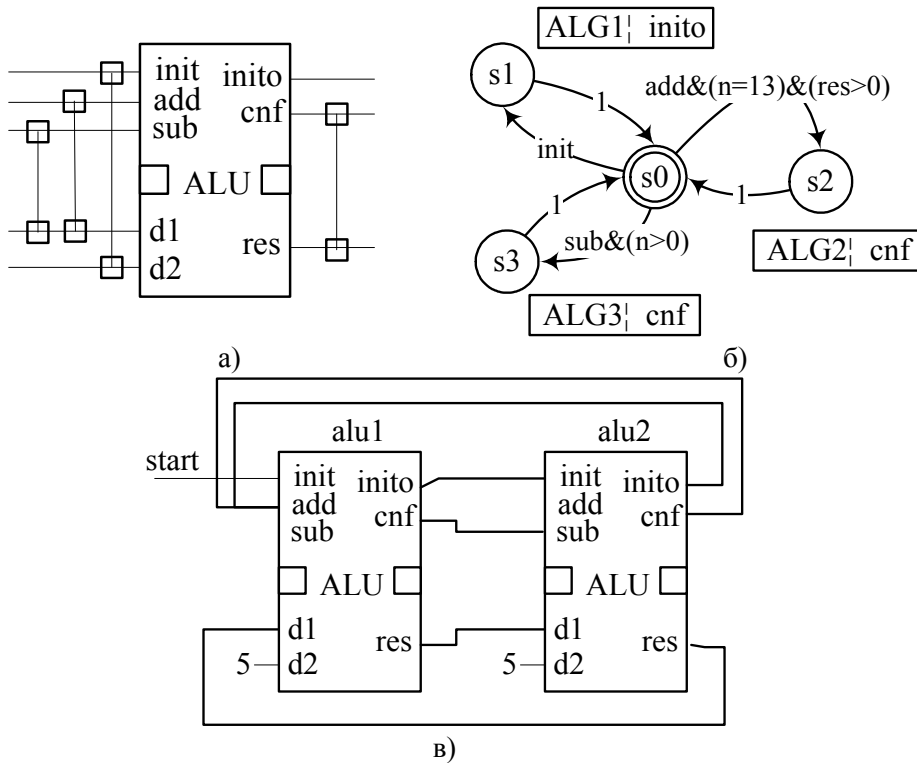


Рис. 1. Пример системы ФБ: а – интерфейс ФБл-типа ALU; б – диаграмма ЕСС для ФБл-типа ALU; в – сеть ФБ

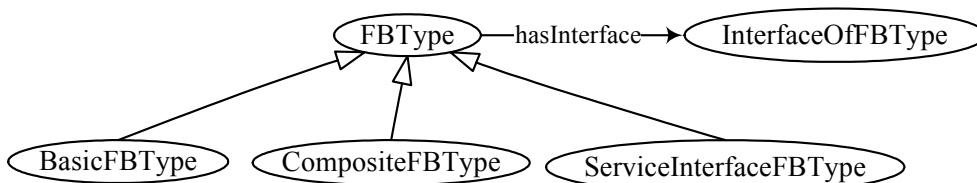


Рис. 2. Типы функциональных блоков

Описание базисных ФБ

Базисные ФБ являются основными в стандарте. Они включают машину состояний, управляемую событиями. Для ее представления используется диаграмма управления выполнением (диаграмма ЕСС). Диаграмма ЕСС состоит из ЕС-состояний и ЕС-переходов. С каждым ЕС-состоянием может быть связан набор ЕС-акций, каждая из которых может включать алгоритм и выходное событие. С ЕС-переходом связывается условие перехода, которое может состоять из входного события и сторожевого условия. Пример базисного ФБл приведен на рис. 1. Семантическая сеть, описывающая тип базисного ФБл, представлена на рис. 4.

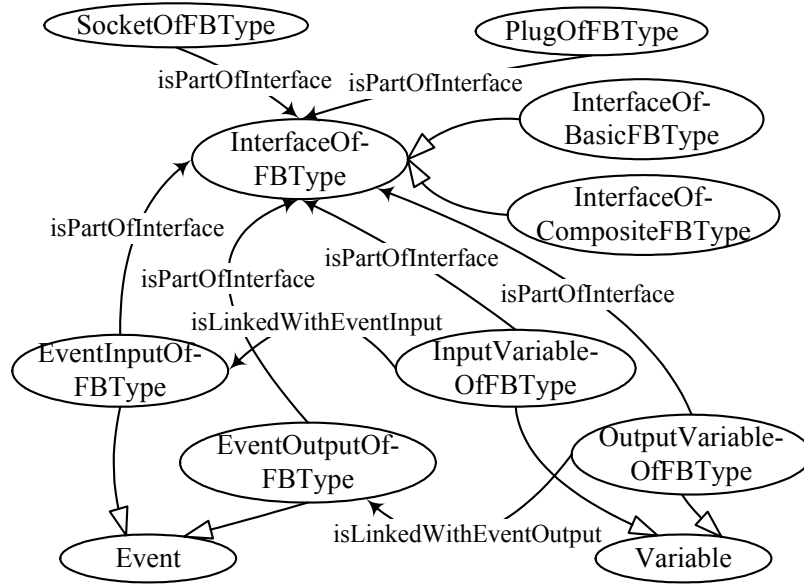


Рис. 3. Определение интерфейсов ФБ-типов

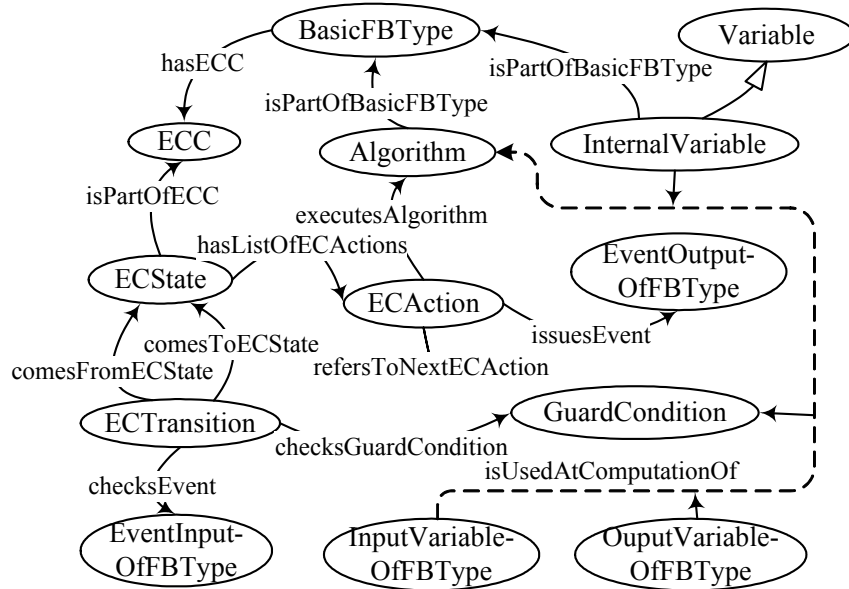


Рис. 4. Определение (типов) базисных ФБ

Описание составных ФБ

Составной ФБл определяется входящей в его состав сетью ФБ, включающей компонентные ФБ и связи между ними. При этом различаются событийные, информационные и адаптерные связи. На рис. 1, в приведен пример сети ФБ, входящей в состав составного ФБ, а на рис. 5 показана семантическая сеть для определения составных ФБ.

Описание конфигураций

Как следует из стандарта ИЕС 61499, устройства и ресурсы, входящие в состав системы, описывают абстрактную архитектуру для выполнения при-

ложений. Приложение представляет собой сеть ФБ для решения определенной задачи в области управления. Состав и взаимосвязь внутренних компонентов системы, устройства или ресурса определяют их конфигурацию. Для использования преимуществ модульного проектирования в стандарте введена типизация на уровне устройств и ресурсов. Ресурсы и устройства могут явно связываться с сегментами вычислительной сети. На рис. 6 приведен пример соотношения приложений, устройств, ресурсов и сегментов.

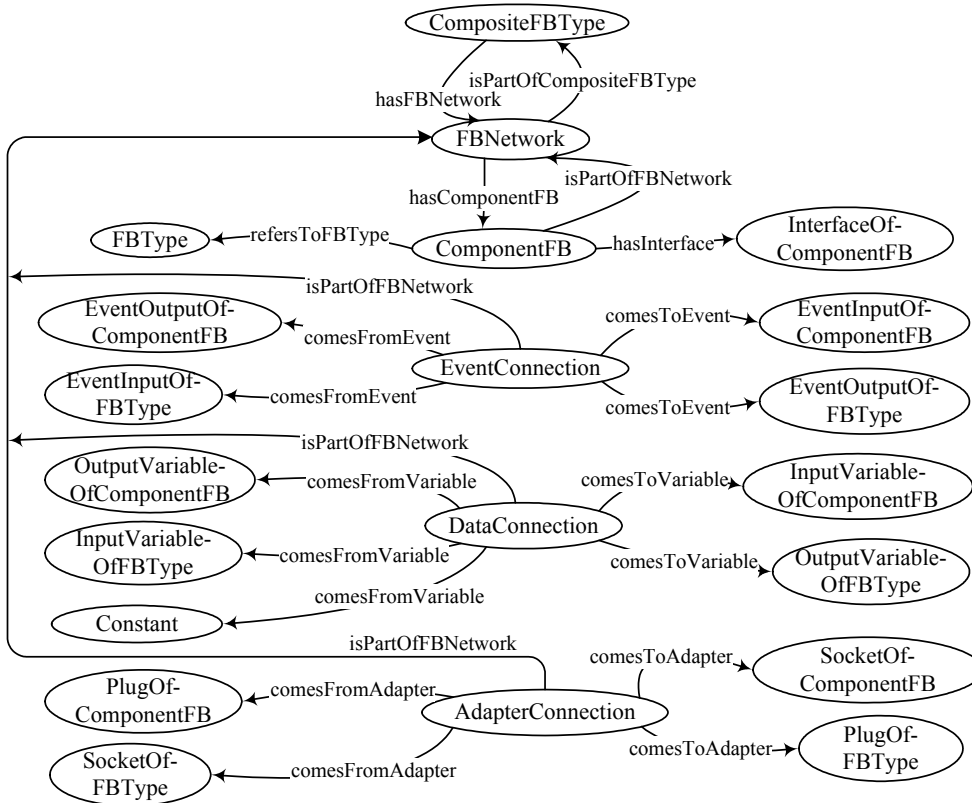


Рис. 5. Определение составных ФБ

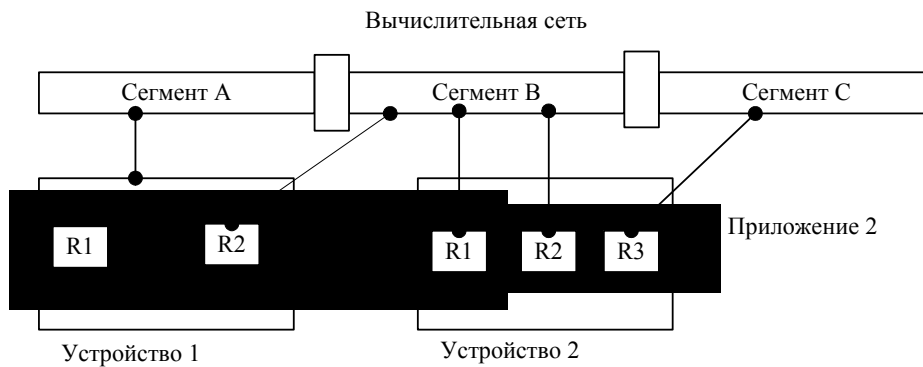


Рис. 6. Иллюстрация системных конфигураций

На рис. 7 изображена семантическая сеть для определения системных конфигураций.

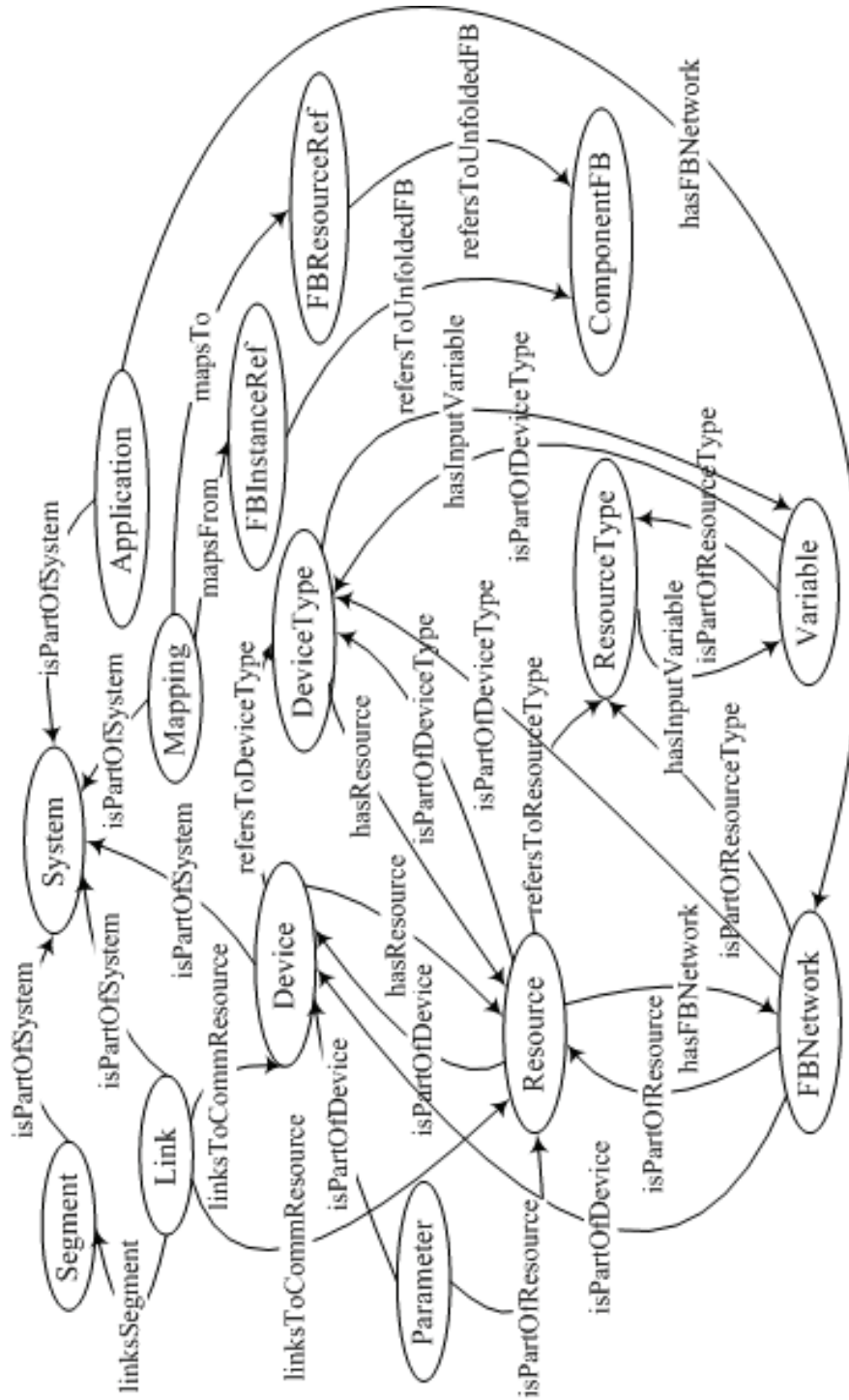


Рис. 7. Определение системных конфигураций

3. Семантический анализ описаний систем ФБ на основе дескриптивной логики

Семантический анализ описаний по своей сути сводится к определению семантически правильных конструкций в множестве синтаксически правильных. Синтаксическая правильность XML-описаний проектов стандарта IEC 61499 определяется соответствующим DTD-блоком [10], а ее проверка поддерживается существующими XML-парсерами.

Рассмотрим в качестве примера представление «синтаксического» знания об информационной связи. Информационная связь описывается в стандарте IEC 61499 с помощью следующего синтаксического правила (в виде РБНФ):

```
data_conn ::=
(fb_instance_name '.' output_variable_name | input_variable_name)
'TO' fb_instance_name '.' input_variable_name ';'
| fb_instance_name '.' output_variable_name 'TO' output_variable_name ';'
```

Для упрощения в приведенном выше правиле data_conn не указаны входные и выходные переменные адаптера-штекера.

Онтологическое описание информационных связей только с помощью рангов и доменов отношений comesFromVariable и comesToVariable (рис. 5) было бы не совсем точным. В этом случае в множество связей DataConnection входила бы связь между входными и выходными переменными оболочки, что не является правильным. Более точно синтаксические знания, заложенные в правиле data_conn, могут быть представлены следующей аксиомой в виде формулы ДЛ типа SROIQ:

$$\text{DataConnection} \sqsubseteq \text{Connection} \sqcap (=1 \text{ comesFromVariable.OutputVariableOfComponentFB} \sqcup =1 \text{ comesFromVariable.InputVariableOfFBType} \sqcup =1 \text{ comesFromVariable.Constant}) \sqcap (=1 \text{ comesToVariable.InputVariableOfComponentFB}) \sqcup (=1 \text{ comesFromVariable.OutputVariableOfComponentFB} \sqcap =1 \text{ comesToVariable.OutputVariableOfFBType})$$

Следует отметить, что в данную формулу введено описание часто используемой на практике связи между константой и входной переменной компонентного ФБл, которое разработчиками стандарта почему-то не было включено в правило data_conn.

Предлагается метод семантического анализа онтологических описаний систем управления на основе ФБ, основанный на использовании определения так называемых «семантически правильных» классов. *Семантически правильный класс* строится на основе соответствующего базового «синтаксически правильного» класса, но при этом в его определение включаются дополнительные семантические свойства и ограничения, полученные из анализа текста стандарта или практики его использования. Сама техника проведения семантического анализа сводится к обычной процедуре классификации, осуществляемой штатной системой рассуждений (reasoner). После проведения классификации производится сравнение множеств экземпляров, отнесенных как к базовому классу, так и к его семантически правильному аналогу. Равенство данных множеств свидетельствует о семантической корректности описания, а неравенство – о наличии ошибки в описании. Локализация ошибки

сводится к поиску в базовом классе экземпляров, не отнесенных к семантически правильному классу, и их дополнительной (ручной) проверке.

Для примера рассмотрим семантически правильный класс `RightInputVariableOfFBType`. Этот класс представляет корректно описанные и корректно использованные входные переменные оболочек (типов) ФБ и определяется следующей аксиомой в виде формулы ДЛ типа SROIQ:

$$\text{RightInputVariableOfFBType} \equiv \text{InputVariableOfFBType} \sqcap (=1 \text{ isPartOfInterface. InterfaceOfBasicFBType} \sqcap \exists \text{ isUsedAtComputationIn. (Algorithm} \sqcup \text{GuardCondition)}) \sqcup (=1 \text{ isPartOfInterface. InterfaceOfCompositeFBType} \sqcap \exists \text{ originatesDataConnection. DataConnection}) \sqcap \exists \text{ isLinkedWithEventInput.EventInput}$$

В соответствии с формульным описанием этого класса: 1) правильная входная переменная оболочки ФБл относится к классу входных переменных оболочек ФБ; 2) переменная должна быть связана WITH-связью хотя бы с одним событийным входом; 3) должно быть верно одно из двух утверждений:

– в случае базисного ФБл она хоть один раз используется в вычислениях каких-либо алгоритмов или сторожевых условий;

– в случае составного ФБл она начинает хотя бы одну информационную дугу.

Для представления корректных событийных входов компонентных ФБ (класс `RightEventInputOfComponentFB`) может использоваться следующая аксиома:

$$\text{RightEventInputOfComponentFB} \equiv \text{EventInputOfComponentFB} \sqcap \exists \text{ finishesEventConnection. EventConnection}$$

Для тестирования разработанной онтологии был создан ряд баз фактов `Abox`, одна из которых представляла систему из двух ФБ типа ALU (рис. 1). При этом экземплярам класса `EventInputOfComponentFB` были даны имена `alu1.init`, `alu1.add`, `alu1.sub`, `alu2.init`, `alu2.add` и `alu2.sub`. В качестве примера ниже приводится описание экземпляра `alu1.add` на языке системы Protégé [11]:

```
isPartOfInterface alu1.Interface
finishesEventConnection alu2.cnf_to_alu1.add_connection
finishesEventConnection alu2.inito_to_alu1.add_connection
```

В результате классификации из всего множества событийных входов компонентных ФБ (в виде экземпляров) к правильным были отнесены все экземпляры, за исключением `alu1.sub` и `alu2.add`, поскольку к соответствующим входам не подведена ни одна событийная связь. В принципе это не является ошибкой, но может служить предупреждением.

4. Семантический анализ описаний систем ФБ на основе правил SWRL

Следует отметить, что не все семантически правильные классы могут быть выражены с помощью только ДЛ. В этом случае для повышения описательных возможностей ДЛ полезно использовать правила языка SWRL [7]. В качестве примера кратко рассмотрим следующее семантическое свойство (положение), явно не прописанное в стандарте, но согласующееся со стандар-

том и используемое на практике. Согласно этому положению при распределении связанных в приложении ФБ на разные ресурсы связь между ними разрывается и взаимодействие этих ФБ производится через коммуникационные ФБ (вид СИФБ) (рис. 8,а). При распределении же ФБ на один и тот же ресурс основным требованием является эквивалентность типов ФБл приложения и соответствующему ему ФБл ресурса (рис. 8,б). При развертывании системы ФБ связь между ФБ приложения мигрирует в связь между соответствующими ФБ ресурса (рис. 8,б).

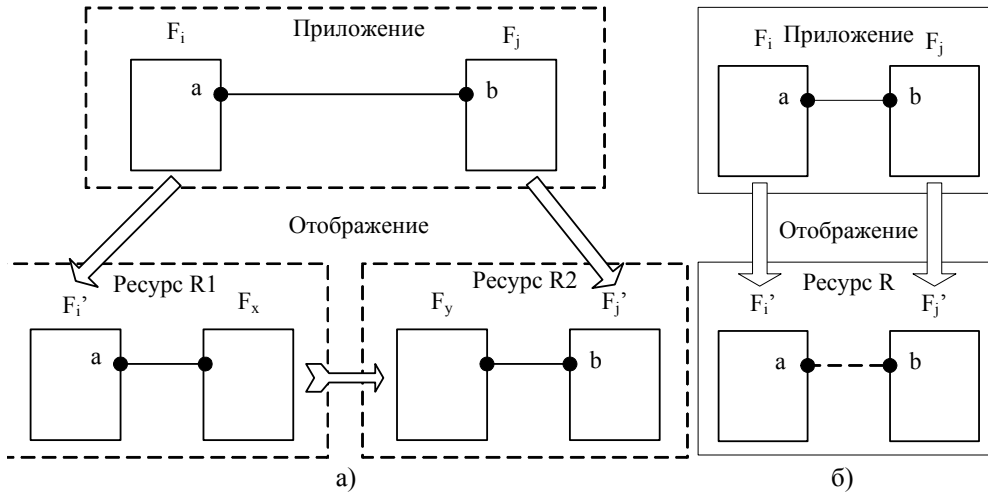


Рис. 8. Распределение функциональных блоков приложения:
 а – на разные ресурсы; б – на один и тот же ресурс

Введем класс `RightAppEventConnection`, описывающий семантически правильные событийные связи в приложении в соответствии с двумя ситуациями, представленными выше:

$$\text{RightAppEventConnection} \equiv \text{RightAppEventConnection1} \sqcup \text{RightAppEventConnection2}$$

Поскольку представление сложных ситуаций в виде правил является в ряде случаев задачей нетривиальной, в качестве промежуточной формы представления знаний предлагается использование графов запросов [9]. Граф запроса наглядно описывает ситуацию в виде некоторой семантической сети, узлами которой являются переменные. Эти переменные, так же как и в правиле SWRL, будем предварять знаком вопроса (?). Дугами в графе запроса являются отношения между соответствующими классами, представляющими интерес. Граф запроса для ситуации, представленной на рис. 8,а, приведен на рис. 9. Ограничения по переменным в этом случае являются $?b1 \neq ?b2$ и $?r1 \neq ?r2$.

Можно считать, что граф запросов представляет антецедент (тело) правила SWRL. Правило SWRL (в виде хорновского дизъюнкта), построенное на основе графа запросов на рис. 9, приведено на рис. 10. Данное правило служит для вычисления класса `RightAppEventConnection1`.

Граф запроса для ситуации о распределении ФБ на один и тот же ресурс, приведен на рис. 11, а соответствующее SWRL-правило – на рис. 12.

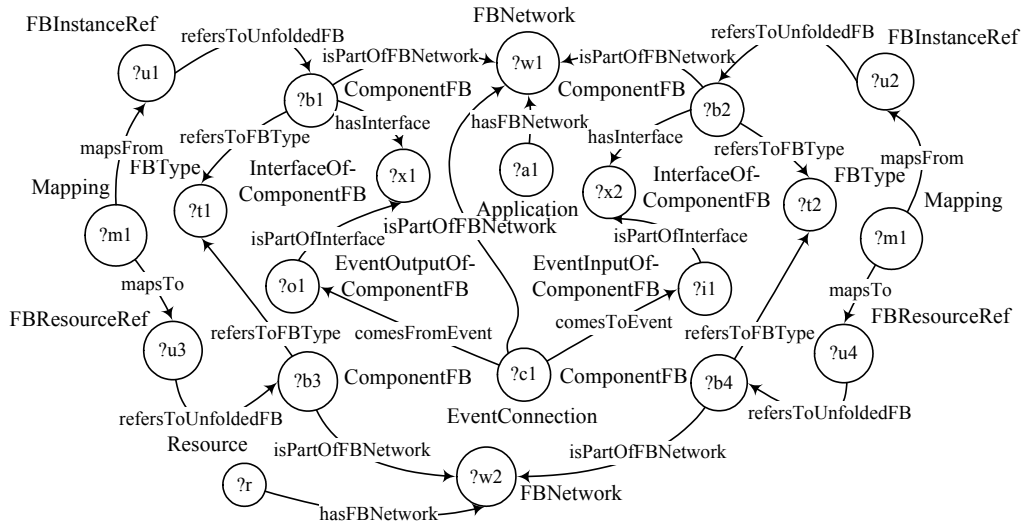


Рис. 11. Граф запроса для ситуации о распределении ФБ на один и тот же ресурс

EventConnection(?c1) \wedge comesFromEvent(?c1,?o1) \wedge comesToEvent(?c1,?i1) \wedge Application(?a1) \wedge hasFBNetwork(?a1,?w1) \wedge isPartOfFBNetwork(?c1,?w1) \wedge isPartOfInterface(?o1,?x1) \wedge hasInterface(?b1,?x1) \wedge isPartOfFBNetwork(?b1,?w1) \wedge refersToFBType(?b1,?t1) \wedge Mapping(?m1) \wedge mapsFrom(?m1,?u1) \wedge mapsTo(?m1,?u3) \wedge refersToUnfoldedFB(?u1,?b1) \wedge refersToUnfoldedFB(?u3,?b3) \wedge refersToFBType(?b3, ?t1) \wedge isPartOfFBNetwork(?b3,?w2) \wedge Resource(?r) \wedge hasFBNetwork(?r,?w2) \wedge isPartOfInterface(?i1,?x2) \wedge hasInterface(?b2,?x2) \wedge isPartOfFBNetwork(?b2,?w1) \wedge refersToFBType(?b2,?t2) \wedge Mapping(?m2) \wedge mapsFBFrom(?m2,?b2) \wedge mapsFBTo(?m2,?b4) \wedge refersToUnfoldedFB(?u2,?b2) \wedge refersToUnfoldedFB(?u4,?b4) \wedge refersToFBType(?b4, ?t2) \wedge isPartOfFBNetwork(?b4,?w2) \rightarrow RightApplEventConnection2(?c1)

Рис. 12. Правило SWRL для вычисления правильных событийных связей в приложении (случай распределения ФБ на один и тот же ресурс)

Следует отметить, что для повышения эффективности вычислений и упрощения отладки (следуя модульному принципу) приведенные правила можно было бы разбить на несколько правил меньшей размерности. Однако в данном случае разбиение правила на модули для иллюстративных целей большого смысла не имеет.

5. Система семантического анализа проектов ИЕС 61499

На основе разработанного выше подхода предлагается структура программных средств для семантического анализа проектов стандарта ИЕС 61499. Основными компонентами системы являются:

- 1) терминологический словарь Tbox, относящийся к стандарту ИЕС 61499;
- 2) транслятор XML-описания ФБ в OWL/SWRL-представление, включающее кроме словаря Tbox и правил SWRL также соответствующую базу фактов Abox;
- 3) интерфейсная программа, осуществляющая взаимодействие с системой рассуждений (ризнер) и пользователем (рис. 13).

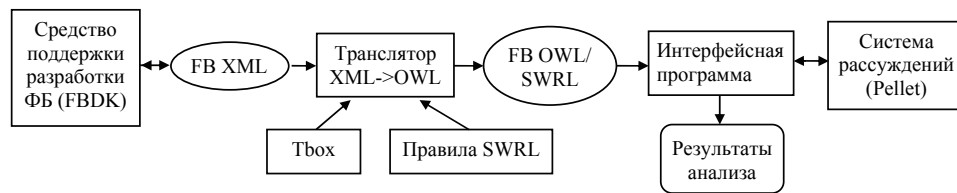


Рис. 13. Система семантического анализа проектов IEC 61499 на основе онтологий

В качестве системы рассуждений можно выбрать свободно распространяемый ризнер Pellet [12]. Он поддерживает ДЛ типа SROIQ и язык SWRL с «ДЛ-безопасной» семантикой. Интерфейсной программой на первый случай может служить система Protégé [11]. Разработка собственной интерфейсной программы с использованием OWL API позволит более эффективно организовать вычисления.

Заключение

В данной работе предложен подход к описанию систем ФБ стандарта IEC 61499 на основе дескриптивной логики и логики хорновских дизъюнктов. На основе предложенного подхода с использованием языков OWL DL и SWRL, а также инструментальной системы Protégé разработана онтология систем управления, построенных в соответствии со стандартом IEC 61499. Данная Web-онтология может считаться разделяемым ресурсом и быть размещена в сети Интернет для общего пользования. Предложенный метод семантического анализа описаний систем управления, построенных в соответствии со стандартом IEC 61499, позволяет выявить семантические ошибки в описании на ранних стадиях проектирования, что сокращает общее время проектирования системы. В работе также предложена структура системы семантического анализа проектов IEC 61499 на основе Web-онтологий.

Направлением дальнейших исследований является расширение онтологии за счет включения в нее описания функционирования систем ФБ с учетом моделей их выполнения.

Список литературы

1. **Гаврилова, Т. А.** Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.
2. International Standard IEC 61499. Function blocks for industrial-process measurement and control systems. Part 1: Architecture / International Electrotechnical Commission. – Geneva, 2005. – 111 p.
3. The Description Logic Handbook: Theory, Implementation, and Applications / F. Baader [et al.]. – Cambridge University Press. – 2003. – 574 p.
4. **Люггер, Дж. Ф.** Искусственный интеллект: стратегии и методы решения сложных проблем : пер. с англ. / Дж. Ф. Люггер. – М. : Изд. дом «Вильямс», 2003. – 864 с.
5. W3C Консорциум. Язык Web-онтологий OWL [Электронный ресурс]. – URL: <http://www.w3.org/2004/OWL/>.
6. **Dubin, V.** Using Prolog for Modelling And Verification of IEC 61499 Function Blocks and Applications / V. Dubin, V. Vyatkin, H.-M. Hanisch // 11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2006), Proceedings. – Prague, Czech Republic, 2006. – P. 774–781.

7. SWRL: A Semantic Web Rule Language. W3C Member Submission, 2004. [Электронный ресурс]. – URL: <http://www.w3.org/Submission/SWRL/>.
8. **Motik, B.** Query answering for OWL DL with rules / B. Motik, U. Sattler, R. Studer // Journal of Web Semantics. – 2005. – № 3 (1). – P. 41–60.
9. Робототехнические комплексы. Искусственный интеллект. Робототехника и гибкие автоматизированные производства : в 9-ти кн. Кн. 6. Техническая имитация интеллекта : учеб. пособие для вузов / В. М. Назаретов, Д. П. Ким ; под ред. И. М. Макарова. – М. : Высшая школа, 1986. – 144 с.
10. International Standard IEC 61499. Function blocks for industrial-process measurement and control systems. Part 2: Software tool requirements / International Electrotechnical Commission. – Geneva, 2005. – 41 p.
11. Protégé [Электронный ресурс]. – URL: <http://protege.stanford.edu>.
12. Pellet – an open-source Java OWL DL reasoner [Электронный ресурс]. – URL: <http://pellet.owldl.com/>.

Дубинин Виктор Николаевич

кандидат технических наук, доцент,
кафедра вычислительной техники,
Пензенский государственный
университет

E-mail: victor_n_dubinin@yahoo.com

Dubinin Viktor Nikolaevich

Candidate of engineering sciences,
associate professor, sub-department
of computer engineering,
Penza State University

Вяткин Валерий Владимирович

доктор технических наук, профессор,
факультет электронных и компьютерных
наук, Оклендский университет
(г. Окленд, Новая Зеландия)

E-mail: victor_n_dubinin@yahoo.com

Vyatkin Valery Vladimirovich

Doctor of engineering sciences, professor,
department of electronics and computer
science, Oakland University
(Oakland, New Zealand)

УДК 004.82: 004.4'2

Дубинин, В. Н.

Семантический анализ описаний систем управления промышленными процессами на основе стандарта IEC 61499 с использованием онтологий / В. Н. Дубинин, В. В. Вяткин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2010. – № 3 (15). – С. 3–15.