УДК 004.8

А.Ф. КУРГАЕВ, С.Н. ГРИГОРЬЕВ

ОПРЕДЕЛЕНИЕ ЯЗЫКОВ XML/RDF SEMANTIC WEB В МЕТАЯЗЫКЕ НОРМАЛЬНЫХ ФОРМ ЗНАНИЙ

Аннотация. Даны формальные текстовые описания проблемно-ориентированных языков XML/RDF — базовых языков Semantic Web. Наличие таких описаний гарантирует реализуемость языков XML/RDF с использованием интерпретатора метаязыка нормальных форм знаний (НФЗ). Показано, что выразительные возможности метаязыка НФЗ для описания синтаксиса этой пары языков вполне сопоставимы с выразительными возможностями EBNF.

Ключевые слова: метаязык нормальных форм знаний, формальное описание метаязыка, eXtensible Markup Language, Resource Description Framework, Semantic Web.

ВВЕДЕНИЕ

Semantic Web является современной реализацией информационно-коммуникативных технологий, цель которой — автоматизация интеллектуальных процессов обработки информации в сети. Semantic Web дает возможность как людям, так и машинам находить, читать, воспринимать и использовать данные из Web для выполнения нужных задач. Semantic Web не является отдельной Web, но продолжением текущей, приданием информации четко определенного смысла, что способствует взаимодействию людей с компьютерами.

Обработкой и обменом информации, выявлением знаний должны заниматься не люди, а интеллектуальные агенты. Для того чтобы агенты могли взаимодействовать между собой и с людьми, в Semantic Web предусмотрено общее (принятое всеми) формальное представление любого ресурса [1].

HTML-страница описывает визуальное представление информации о Web-ресурсе и трудно поддается смысловому анализу компьютерами. Semantic Web разделяет средства визуализации (HTML) и средства смыслового содержания (XML $\rightarrow$ RDF + OWL).

Semantic Web создает общую Web-структуру, в которой различаются данные (факты) и метаданные (знания о данных), многократно используемые не только в Internet, но и в разнообразных приложениях для решения актуальных задач. Описание знаний в Semantic Web реализуется на языке Resource Description Framework (RDF) в форме определений терминов предметной области и на языке онтологий Web Ontology Language (OWL), описывающем семантическую взаимосвязь ресурсов на RDF. Эти языки используют eXtensible Markup Language (XML) для описания синтаксиса и Uniform Resource Identifier (URI) — для идентификации всевозможных ресурсов. Таким образом, XML позволяет пользователям добавлять в свои документы произвольную структуру, не вникая в ее смысл [2].

Semantic Web создается на основе ряда стандартов, рекомендуемых консорциумом W3C, и инструментальных средств.

Цель статьи — описание определения языков XML/RDF — базовых языков Semantic Web в метаязыке нормальных форм знаний (НФЗ) [3–6] для эмпирической апробации выразительных возможностей метаязыка НФЗ.

Используем последнюю, доступную авторам официальную редакцию нормативных документов языка XML [7], принятую консорциумом W3C.

Формальное определение XML в метаязыке НФЗ [3–6]. Формальное определение языка XML, описанное в [7] метаязыком Extended Backus–Naur Form (EBNF), в нотации метаязыка НФЗ представлено системой правил, изложенных ниже.

Согласно [7] текст является правильно сформированным XML-документом, если он соответствует продукции, определяющей термин document.

```
/* любой правильно сформированный XML-документ /*
 (1)   document          = prolog element (Misc);
/* Символы документа: любой символ Unicode, без суррогатных блоков FFFE
и FFFF/*
 (2)   Char              = Char1 / Char2 / Char3/ Char4;
/* Пустое пространство из: 0009 – горизонтальная табуляция (HT), 000A —
перевод строки (LF), 000D — возврат каретки (CR), 0020 — пробел (SP) /*
 (3)   S                 = S1 (S1);
 (3a)  S1                = '#x20' / '#x9' / '#xD' / '#xA';
/* Имена и лексемы /*
 (4)   NameStartChar     = ': ' / Letter1 / '_' / Letter2 / NSC1 / NSC2 / NSC3 /
                           NSC4 / NSC5 / NSC6 / NSC7 / NSC8 / NSC9 /
                           NSC10 / NSC11 / NSC12;
 (4a)  NameChar          = NameStartChar / '-' / '.' / Numeral / '#xB7' / NC1 / NC2;
 (5)   Name              = NameStartChar (NameChar);
 (6)   Names             = Name ('#x20' Name);
 (7)   Nmtoken           = NameChar (NameChar);
 (8)   Nmtokens          = Nmtoken ('#x20' Nmtoken);
/* Литералы /*
 (9)   EntityValue       = '"' (^Ch1 Char / PEReference / Reference) '"' / "'"
                           (^Ch2 Char / PEReference / Reference) "'";
(10)   AttValue          = '"' (^Ch3 Char | Reference) '"' / "'" (^Ch4 | Reference) "'";
(11)   SystemLiteral     = '"' (^" Char) '"' / "'" (^' Char) "'";
(12)   PubidLiteral      = '"' (PubidChar) '"' / "'" (^"'" PubidChar) "'";
(13)   PubidChar         = '#x20' / '#xD' / '#xA' / Letter1 / Letter2 / Numeral / PC;
/* Знаки данных /*
(14)   CharData          = ^CharData1 (^Ch5 Char);
(14a)  CharData1         = (Ch5 Char)>']]' (^'<&' Char);
/* Комментарий /*
(15)   Comment           = '<!--' ( ^'-' Char / '-' ^'-' Char) '-->';
/* Инструкции по обработке /*
(16)   PI                = '<?' PITarget PI1 '?>';
(16a)  PI1               = S ^PI2 (Char) / true;
(16b)  PI2               = (Char) '?>' (Char);
(17)   PITarget          = ^PITarget_ Name;
(17a)  PITarget_         = PitX PitM PitL;
```

```
(17b)  PitX          = 'X' | 'x';
(17c)  PitM          = 'M' | 'm';
(17d)  PitL          = 'L' | 'l';

/* Разделы CDATA /*

(18)   CDSect        = CDStart CData CDEnd;
(19)   CDStart       = '<![CDATA[';
(20)   CData         = ^Cdata1 (Char);
(20a)  Cdata1        = (Char) ']]>' (Char);
(21)   CDEnd         = ']]>';

/* Пролог: объявление типа документа /*

(22)   prolog        = XMLDecl (Misc) docMisc;
(23)   XMLDecl       = '<?xml' VersionInfo EncodingDecl_ SDDecl_ S_ '?>' / true;
(23a)  EncodingDecl_ = EncodingDecl / true;
(23b)  SDDecl_       = SDDecl / true;
(23c)  S_            = S / true;
(23d)  docMisc       = doctypedecl (Misc) / true;
(24)   VersionInfo   = S 'version' Eq VerNum;
(24a)  VerNum        = '"' VersionNum '"' / "'" VersionNum "'";
(25)   Eq            = S_ '=' S_;
(26)   VersionNum    = '1.' Numeral (Numeral);
(27)   Misc          = Comment / PI / S;

/* Определение типа документа /*

(28)   doctypedecl   = '<!DOCTYPE' S Name SEID S_ iSS '>';
(28a)  DeclSep       = PEReference / S;
(28b)  intSubset     = (markupdecl / DeclSep);
(28c)  SEID          = S ExternalID / true;
(28d)  iSS           = '[' intSubset ']' S_ / true;
(29)   markupdecl    = elementdecl / AttlistDecl / EntityDecl / NotationDecl /
                       PI / Comment;

/* Внешнее подмножество /*

(30)   extSubset     = TextDecl_ extSubsetDecl;
(30a)  TextDecl_     = TextDecl / true;
(31)   extSubsetDecl = ( markupdecl / conditionalSect / DeclSep );

/* Декларация одиночного документа /*

(32)   SDDecl        = S 'standalone' Eq SDDyesno1;
(32a)  SDDyesno1     = '"' SDDyesno2 '"' / "'" SDDyesno2 "'";
(32b)  SDDyesno2     = 'yes' / 'no';

/* Элемент /*

(39)   element       = EmptyElemTag / STag content Etag;

/* Старт-тэг /*

(40)   STag          = '<' Name (S Attribute) S_ '>';
(41)   Attribute     = Name Eq AttValue;

/* Оконечный тэг /*

(42)   ETag          = '</' Name S_ '>'

/* Содержание элементов /*
```

```
(43)   content          =   CharData_ (content_ CharData_ );
(43a)  content_         = element / Reference / CDSect / PI / Comment;
(43b)  CharData_        = CharData / true;

/* Метки для пустых элементов /*

(44)   EmptyElemTag     = '<' Name (S Attribute) S_ '/>';

/* Декларация типа элемента /*

(45)   elementdecl      = '<!ELEMENT ' S Name S contentspec S_ '>';
(46)   contentspec      = 'EMPTY' / 'ANY' / Mixed / children;

/* Модель контента элемента /*

(47)   children         = choiseq children_;
(47a)  choiseq          = choice / seq ;
(47b)  children_        = '?' / '*' / '+' / true;
(48)   cp               = cp_ children_;
(48a)  cp_              = Name / choice / seq;
(49)   choice           = '(' S_ cp S_ '/' S_ cp (S_ '/' S_ cp ) S_ ')';
(50)   seq              = '(' S_ cp ( S_ ',' S_ cp ) S_ ')';

/* Декларация смешанного содержимого /*

(51)   Mixed            = '(' S_ '#PCDATA' (S_ '/' S_ Name) S_ ')*'
                        / '(' S_ '#PCDATA' S_ ')';

/* Декларация списка атрибутов /*

(52)   AttlistDecl      = '<!ATTLIST ' S Name (AttDef) S_ '>';
(53)   AttDef           = S Name S AttType S DefaultDecl;

/* Типы атрибутов /*

(54)   AttType          = StringType / TokenizedType / EnumeratedType;
(55)   StringType       = 'CDATA';
(56)   TokenizedType    = 'ID' / 'IDREF' / 'IDREFS' / 'ENTITY' / 'ENTITIES'
                          / 'NMTOKEN' / 'NMTOKENS';

/* Перечисляемые типы атрибутов /*

(57)   EnumeratedType   = NotationType / Enumeration;
(58)   NotationType     = 'NOTATION' S '(' S_ Name (S_ '/' S_ Name) S_ ')';
(59)   Enumeration      = '(' S_ Nmtoken (S_ '/' S_ Nmtoken) S_ ')';

/* Атрибут по умолчанию /*

(60)   DefaultDecl      = '#REQUIRED' / '#IMPLIED' / DefaultDecl_ AttValue;
(60a)  DefaultDecl_     = '#FIXED' S / true;

/* Условный раздел /*

(61)   conditionalSect  = includeSect / ignoreSect;
(62)   includeSect      = '<![' S_ 'INCLUDE' S_ '[' extSubsetDecl ']]>';
(63)   ignoreSect       = '<![' S_ 'IGNORE' S_ '[' (ignoreSectContents) ']]>';
(64)   ignoreSectContents = Ignore ('<![' ignoreSectContents ']]>' Ignore);
(65)   Ignore           = ^Ignore1 (Char);
(65a)  Ignore1          = (Char) Ignore2 (Char);
(65b)  Ignore2          = '<![' / ']]>';
```

```
/* Ссылка на символ /*
(66)   CharRef          = '&#' Numeral (Numeral) ';' / '&#x'
                          CharRef_ (CharRef_) ';' ;
(66a)  CharRef_         = Numeral / a / b / c / d / e / f / A / B / C / D / E / F;
/* Ссылка на сущность /*
(67)   Reference        = EntityRef / CharRef;
(68)   EntityRef        = '&' Name ';' ;
(69)   PEReference      = '%' Name ';' ;
/* Декларация сущности /*
(70)   EntityDecl       = GEDecl / PEDecl;
(71)   GEDecl           = '<!ENTITY' S Name S EntityDef S_ '>';
(72)   PEDecl           = '<!ENTITY' S '%' S Name S PEDef S_ '>';
(73)   EntityDef        = EntityValue / ExternalID NdataDecl_;
(73a)  NdataDecl_       = NdataDecl / true;
(74)   PEDef            = EntityValue / ExternalID;
/* Декларация внешней сущности /*
(75)   ExternalID       = 'SYSTEM' S SystemLiteral / 'PUBLIC' S PubidLiteral S
                          SystemLiteral;
(76)   NDataDecl        = S 'NDATA' S Name;
/* Декларация текста /*
(77)   TextDecl         = '<?xml' VersionInfo_ EncodingDecl S_ '?>';
(77a)  VersionInfo_     = VersionInfo / true;
/* Внешняя сущность правильно сформирована /*
(78)   extParsedEnt     = TextDecl_ content;
(78a)  TextDecl_        = TextDecl / true;
/* Кодирование декларации (содержит только латинские символы) /*
(80)   EncodingDecl     = S 'encoding' Eq EncodingDecl_;
(80a)  EncodingDecl_    = '"' EncName '"' / "'" EncName "'" ;
(81)   EncName          = EncName_ (EncName_ / Numeral / '-');
(81a)  EncName_         = Letter1 / Letter2;
/* Нотация декларации /*
(82)   NotationDecl     = '<!NOTATION' S Name S NotationDecl_ S_ '>';
(82a)  NotationDecl_    = ExternalID / PublicID;
(83)   PublicID         = 'PUBLIC' S PubidLiteral;
/* Терминалы /*
(T1)   Char1            ~ #x9 | #xA | #xD;
(T2)   Char2            ~ [#x20-#xD7FF];
(T3)   Char3            ~ [#xE000-#xFFFD];
(T4)   Char4            ~ [#x10000-#x10FFFF];
(T5)   Ch1              ~ [^%&"];
(T6)   Ch2              ~ [^%&'];
(T7)   Ch3              ~ [^<&"];
(T8)   Ch4              ~ [^<&'];
(T9)   Ch5              ~ [^<&];
(T10)  Letter1          ~ [A-Z];
```

(T11) Letter2 ~ [a-z];
(T12) NSC1 ~ [#xC0-#xD6];
(T13) NSC2 ~ [#xD8-#xF6];
(T14) NSC3 ~ [#xF8-#x2FF];
(T15) NSC4 ~ [#x370-#x37D];
(T16) NSC5 ~ [#x37F-#x1FFF];
(T17) NSC6 ~ [#x200C-#x200D];
(T18) NSC7 ~ [#x2070-#x218F];
(T19) NSC8 ~ [#x2C00-#x2FEF];
(T20) NSC9 ~ [#x3001-#xD7FF];
(T21) NSC10 ~ [#xF900-#xFDCF];
(T22) NSC11 ~ [#xFDF0-#xFFFD];
(T23) NSC12 ~ [#x10000-#xEFFFF];
(T24) Numeral ~ [0-9];
(T25) NC1 ~ [#x0300-#x036F];
(T26) NC2 ~ [#x203F-#x2040];
(T27) PC ~ [- ' ()+,./ :=?;!*#@$_%];

В этом описании использованы следующие метасимволы метаязыка НФЗ [3–6]:

= — разделитель, отделяет имя понятия (нетерминала) от его определения;

; — конец определения понятия;

" " (пробел) — отношение конкатенации;

/ — отношение альтернативного выбора;

( ) — итерационные скобки обрамляют повторяемую (нуль или больше раз) структуру понятий;

^ — отношение отрицания последующего понятия;

' — текстовая кавычка;

true — тождественно истинное понятие с пустым объемом;

~ — разделитель, отделяет имя терминала от его определения.

Кроме того, в описании семантики терминалов использована нотация из [7] для описания сопоставления им одного или нескольких символов.

Оценка описания определения XML в метаязыке НФЗ. Исходное описание языка XML метаязыком EBNF [7] включает 79 правил, а описание метаязыком НФЗ, представленном выше, включает 113 правил определения нетерминалов и 27 определений терминалов. При этом все терминалы НФЗ-описания, по сути, являются явной формой семантических процедур, представленных и в EBNF-описании, но в неявной форме. Дополнительные (относительно EBNF-описания) правила НФЗ-описания нетерминалов использованы для моделирования:

структурных скобок: правила (3a), (17b), (17c), (17d), (24a), (32a), (32b), (43a), (47a), (48a), (60a), (65b), (66a), (80a), (81a), (82a);

необязательности: правила (16a), (23a), (23b), (23c), (23d), (28c), (28d), (30a), (43b), (47b), (73a), (77a), (78a);

отрицания понятий: правила (14a), (16b), (17a), (20a), (65a).

Основным результатом сопоставления представленного описания с известным является следующее утверждение.

Утверждение 1. Выразительных возможностей метаязыка НФЗ достаточно для описания проблемно-ориентированного метаязыка XML.

ОПИСАНИЕ ОПРЕДЕЛЕНИЯ ЯЗЫКА RDF

Язык RDF является домен нейтральной моделью представления метаданных и их транспортировки для совместного использования независимо разработан-

ных описаний информации в какой-либо области. Цель RDF — описание ресурсов без каких-либо предположений о конкретной предметной области, без определения (априори) семантики любого домена приложения. Синтаксис языка RDF использует XML для задания, кодирования, транспортировки и хранения произвольной семантики в единой форме. В этом отношении RDF и XML являются взаимодополняющими.

Формальное определение RDF в метаязыке НФЗ [3–6]. При составлении в метаязыке НФЗ описания RDF в качестве исходного образца использована официальная редакция нормативных документов модели и спецификации синтаксиса языка RDF, принятая консорциумом W3C [8]. Следует заметить, что синтаксис в [8] является одним из возможных синтаксисов RDF и уже существуют альтернативные способы представления той же модели данных RDF [9].

В нотации метаязыка НФЗ формальное определение языка RDF описывается следующей системой правил.

/* Описание RDF в метаязыке НФЗ /*

```
(1)   RDF            = RDF_st (obj) RDF_end;
(1a)  RDF_st         = '<rdf:RDF>' / true;
(1b)  RDF_end        = '</rdf:RDF>' / true;
(2)   obj            = description | container
(3)   description    = '<rdf:Description' idAboutAttr bagIdAttr (propAttr) '/>'
                       / '<rdf:Description' idAboutAttr bagIdAttr (propAttr) '>'
                       (propertyElt) '</rdf:Description>' / typedNode;
(4)   container      = sequence / bag / alternative;
(5)   idAboutAttr    = idAttr / aboutAttr / aboutEachAttr / true;
(6)   idAttr         = 'ID=" ' IDsymbol ' " ' ;
(6a)  idAttr_        = idAttr / true;
(7)   aboutAttr      = 'about=" ' URI-reference ' " ' ;
(8)   aboutEachAttr  = ' aboutEach=" ' URI-reference ' " ' / ' aboutEachPrefix="
                       ' string' " ' ;
(9)   bagIdAttr      = ' bagID=" ' Idsymbol ' " ' / true;
(10)  propAttr       = typeAttr / propName '=" ' string' " '; /* with embedded
                       quotes escaped */
(11)  typeAttr       = ' type=" ' URI-reference ' " ' ;
(12)  propertyElt    = '<' propName idAttr_ '>' value '</' propName '>'
                       / '<' propName idAttr_ parseLiteral '>' literal '</'
                       propName '>' / '<' propName idAttr_ parseResource '>'
                       (propertyElt) '</' propName '>' / '<' propName
                       idRefAttr_ bagIdAttr (propAttr) '/>' ;
(13)  typedNode      = '<' typeName idAboutAttr bagIdAttr (propAttr) '/>'
                       / '<' typeName idAboutAttr bagIdAttr? (propAttr) '>'
                       (propertyElt) '</' typeName '>' ;
(14)  propName       = Qname;
(15)  typeName       = Qname;
(16)  idRefAttr      = idAttr / resourceAttr / true ;
(17)  value          = obj / string ;
(18)  resourceAttr   = 'resource=" ' URI-reference ' " ' ;
(19)  Qname          = NSprefix_ name ;
(20)  URI-reference  = /* [10] */
(21)  IDsymbol       = NameChar ; /* в описании XML */
(22)  name           = Name ; /* в описании XML */
```

(23) NSprefix = 'xmlns:' ; /* **or** any legal XML namespace prefix */
(23a) NSprefix_ = NSprefix / true ;
(24) string = /* any XML text, with "<", ">", and "&" escaped */
(25) sequence = '<*rdf*:Seq' idAttr_ '>' (member) '</*rdf*:Seq'
/ '<*rdf*:Seq' idAttr_ (memberAttr) '/>' ;
(26) bag = '<*rdf*:Bag' idAttr_ '>' (member) '</*rdf*:Bag'
/ '<*rdf*:Bag' idAttr_ (memberAttr) '/>' ;
(27) alternative = '<*rdf*:Alt' idAttr_ '>' member (member) '</*rdf*:Alt'
/ '<*rdf*:Alt' idAttr_ memberAttr_ '/>' ;
(28) member = referencedItem / inlineItem ;
(29) referencedItem = '<*rdf*:li' resourceAttr '/>' ;
(30) inlineItem = '<*rdf*:li' '>' value </*rdf*:li' / '<*rdf*:li' parseLiteral '>'
literal </*rdf*:li>' / '<*rdf*:li' parseResource '>'
propertyElt* </*rdf*:li' ;
(31) memberAttr = ' *rdf*:_ *n* =" ' string ' " ' ; (where *n* is an integer)
(31a) *n* = integer;
(31b) integer = digit (digit) ;
(31c) digit = /* [10] */
(31d) memberAttr_ = memberAttr / true;
(32) parseLiteral = ' parseType="Literal" ' ;
(33) parseResource = ' parseType="Resource" ' ;
(34) literal = document ; /* в описании XML */

Сопоставление двух описаний определения RDF. Исходное описание в [8] языка RDF метаязыком EBNF включает 34 правила, а описание метаязыком НФЗ — 43 правила определения нетерминалов. Дополнительные правила описания метаязыком НФЗ нетерминалов использованы для моделирования:

необязательности: правила (1a), (1b), (6a), (23a) и (31d);

определения в явной форме понятия *n*: правила (31a), (31b), (31c).

Основным результатом сопоставления этих двух описаний является вывод о достаточных выразительных возможностях метаязыка НФЗ для описания проблемно-ориентированного языка RDF.

ЗАКЛЮЧЕНИЕ

Представленная формализация проблемно-ориентированных языков XML/RDF — базовых языков Semantic Web свидетельствует о сопоставимости выразительных возможностей метаязыка НФЗ с выразительными возможностями EBNF. Эта формализация практически пригодна в качестве спецификации для реализации этих языков в составе других систем.

СПИСОК ЛИТЕРАТУРЫ

1. Berners-Lee T., Hendler J., Lassila O. The Semantic Web: May 2001; Scientific American. URL: http://www.scientificamerican.com/article/the-semantic-web/.
2. Bosak J., Bray T. XML and the Second Generation Web: May 1999; Scientific American. URL: https://courses.washington.edu/infx598/win12/secondGenerationWeb.pdf.
3. Спосіб представлення і використання знань / О.П. Кургаєв, С.М. Григор'єв / Патент на корисну модель UA 92484 U, 2014р., Бюл. №16.
4. Kurgaev A., Grygoryev S. The normal forms of knowledge. *Dopov. NAN Ukraine*. 2015. N 11. P. 36–43.

5. Кургаев А.Ф., Григорьев С.Н. Метаязык нормальных форм знаний. *Кибернетика и системный анализ.* 2016. Т. 52, № 6. С. 11–20.
6. Kurgaev A., Grygoryev S. The universal Turing machine interpreter. *Dopov. NAN Ukraine*. 2016, N 10. P. 30–36. DOI: http://dx.doi.org/10.15407/dopovidi2016.10.030.
7. Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation 26 November 2008. URL: http://www.w3.org/TR/2008/REC-xml-20081126.
8. Status for Resource Description Framework (RDF) Model and Syntax Specification. — W3C Recommendation 22 February 1999. URL: http://w3.org/TR/1999/REC-rdf-syntax-19990222.
9. RDF 1.1 XML Syntax. W3C Recommendation 25 February 2014. URL: http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/.
10. Uniform Resource Identifiers (URI): Generic Syntax. Berners-Lee T. (MIT/LCS), Fielding R., Irvine U.C., Masinter L. (Xerox Corporation). August 1998. URL: http://www.isi.edu/in-notes/rfc2396.txt.

Надійшла до редакції 09.09.2016

О.П. Кургаєв, С.М. Григор'єв

ВИЗНАЧЕННЯ МОВ XML/RDF SEMANTIC WEB У МЕТАМОВІ НОРМАЛЬНИХ ФОРМ ЗНАНЬ

Анотація. Наведено формальні текстові описи проблемно-орієнтованих мов XML/RDF — базових мов Semantic Web. Наявність таких описів гарантує реалізовність мов XML/RDF з використанням інтерпретатора метамови нормальних форм знань (НФЗ). Показано, що виразні можливості метамови НФЗ для опису синтаксису цієї пари мов цілком порівнянні з виразними можливостями EBNF.

Ключові слова: метамова нормальних форм знань, формальний опис метамови, eXtensible Markup Language, Resource Description Framework, Semantic Web.

A.F. Kurgaev, S.N. Grigoriev

DEFINITION OF XML/RDF SEMANTIC WEB LANGUAGES IN THE NORMAL FORMS OF KNOWLEDGE METALANGUAGE

Abstract. Provided are the formal text descriptions of the problem-oriented XML/RDF languages, which are the basic Semantic Web languages. The availability of such descriptions guarantees the ability to implement the XML/RDF languages with the implementation of the interpreter of the NFK metalanguage. It is shown that the expressiveness of the NFK metalanguage in describing the syntax of the given pair of languages is quite comparable with EBNF expressiveness.

Keywords: Normal Forms of Knowledge Metalanguage, formal description of Metalanguage, eXtensible Markup Language, Resource Description Framework, Semantic Web.

Кургаев Александр Филиппович,
доктор техн. наук, профессор, ведущий научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: afkurgaev@ukr.net.

Григорьев Сергей Николаевич,
соискатель Института кибернетики им. В.М. Глушкова НАН Украины, Киев,
e-mail: Sergey@Grigoriev.kiev.ua.