ЭТАП РАЗРАБОТКИ ПОЛНОФУНКЦИОНАЛЬНОГО СЕРВЕРА ПРОТОКОЛА Z39.50: КОМПИЛЯЦИЯ ПОДМНОЖЕСТВА ASN.1 В СТРУКТУРЫ ЯЗЫКА REBOL

Капустин Виктор Андреевич
Междисциплинарный центр Санкт-Петербургского государственного университета
199178 Санкт-Петербург, 14-я линия Васильевского острова, д. 29
vak@icape.nw.ru

A STEP IN Z39.50 SERVER DEVELOPMENT: COMPILING AN ASN.1 SUBSET TO REBOL DATA STRUCTURES.

Victor Kapustin
Interdisciplinary Center for Advanced Professional Education
of Saint-Petersburg State University
vak@icape.nw.ru

An important stage of Z39.50 application development process – compiling ASN.1 protocol description into implementation programming language data structures is considered. I choose REBOL as the target programming language. REBOL basic data structure – block – is syntactically similar to ASN type definition. Z39.50 standard uses the simplest ASN types only, so instead of a real compiler a regular expression transform appeared to be sufficient. In parallel to generation of data structures, all the protocol automata are spawned as a by-product. The development documentation is at http://www.z.nw.ru.

Полнофункциональный сервер протокола Z39.50 должен соответствовать текущему стандарту протокола[1, 2] и допускать использование экспериментальных наборов атрибутов данных. Формальная часть стандарта протокола Z39.50 написана на языке описания абстрактных синтаксисов данных – ASN.1[3]. На этом же языке описываются наборы атрибутов данных[1, 2]. Реализация Z39.50 (как серверов, так и клиентов) требует преобразования этих описаний в конкретный синтаксис языка реализации.

Выбор ASN.1 в качестве средства описания протокола Z39.50 был продиктован необходимостью иметь именно описание так называемого "абстрактного синтаксиса" протокола, совершенно не зависящего от реализации. В середине 80-х годов прошлого века единственным таким средством был ASN.1 (сейчас сравнимой с ASN.1 выразительной силой обладают языки IDL и XML).

В конкретной реализации приложения, использующего протокол, требуется выполнить два преобразования абстрактного синтаксиса прото-

кола (рис. 1): одно – в конкретный синтаксис языка (программирования) реализации для создания структур данных, передаваемых с помощью протокола, и используемых протокольными автоматами, а второе – в синтаксис передачи, или, другими словами, в коммуникационный формат протокола (тот формат данных, который в действительности используют приложения для коммуникации).

В качестве коммуникационного формата Z39.50 использует так называемую BER-кодировку [4], алгоритмы порождения и разбора которой несложны при наличии в программе соответствующих структур данных (на самом деле эти алгоритмы не зависят от структуры передаваемых данных). Реализации этих алгоритмов в REBOL будет посвящена отдельная публикация.

Основной проблемой при создании приложения протокола является эффективное порождение структур данных языка реализации приложения, строго соответствующих формальному описанию протокола на ASN.1. При этом возможны два подхода[5-7]. Первый заключается в ручном преобразовании формального описания абстрактного синтаксиса Z39.50 в конкретный синтаксис языка реализации. Второй – в использовании так называемого ASN-компилятора, который способен автоматически преобразовать текст набора ASN-модулей, представляющих собой описание протокола, в подходящие структуры языка программирования (поскольку такие компиляторы не порождают исполняемого кода, то они часто называются stub-компиляторами – компиляторами заглушек). В программных продуктах Z39.50 с открытым кодом [8-10, 11, 12] преобладает первый подход. Наиболее продвинутый такой продукт – YAZ [13] – использует, однако, ASN- компилятор, который с YAZ не поставляется. В результате в состав YAZ и Z-сервера на его основе – Zebra[14] – входит множество компонентов, которые авторы не рекомендуют изменять. Однако в сам стандарт Z39.50 и в его ASN-описание ежегодно вносятся изменения (например, для устранения обнаруженных противоречивостей).

Как же поступить при разработке полнофункционального Z-сервера? Ручное преобразование ASN-модулей описания Z39.50 чревато ошибками: это описание[1,2] содержит около 2000 строк текста (без учета комментариев), которые представляют собой 220 правил (определений типов)[2] (на самом деле [2] содержит еще и заметное количество опечаток). Поэтому было принято решение использовать ASN-компилятор (или аналогичное компилятору технологическое решение). Существует множество ASN-компиляторов в различные языки программирования [15], однако функциональность бесплатных ASN-компиляторов ограничена. К тому же, нами в качестве языка реализации был выбран свободно распространяемый REBOL [16], основными преимуществами которого, по нашему мнению, являются высокая эффективность, переносимость (REBOL доступен для 59 платформ, исходный код программ на REBOL при этом не требует перера-

ботки) и наглядность текста программ. REBOL, однако, обладает необычным синтаксисом, который, с одной стороны, делает бесполезными существующие ASN-компиляторы [17], но, с другой стороны, значительно облегчает создание такого компилятора.

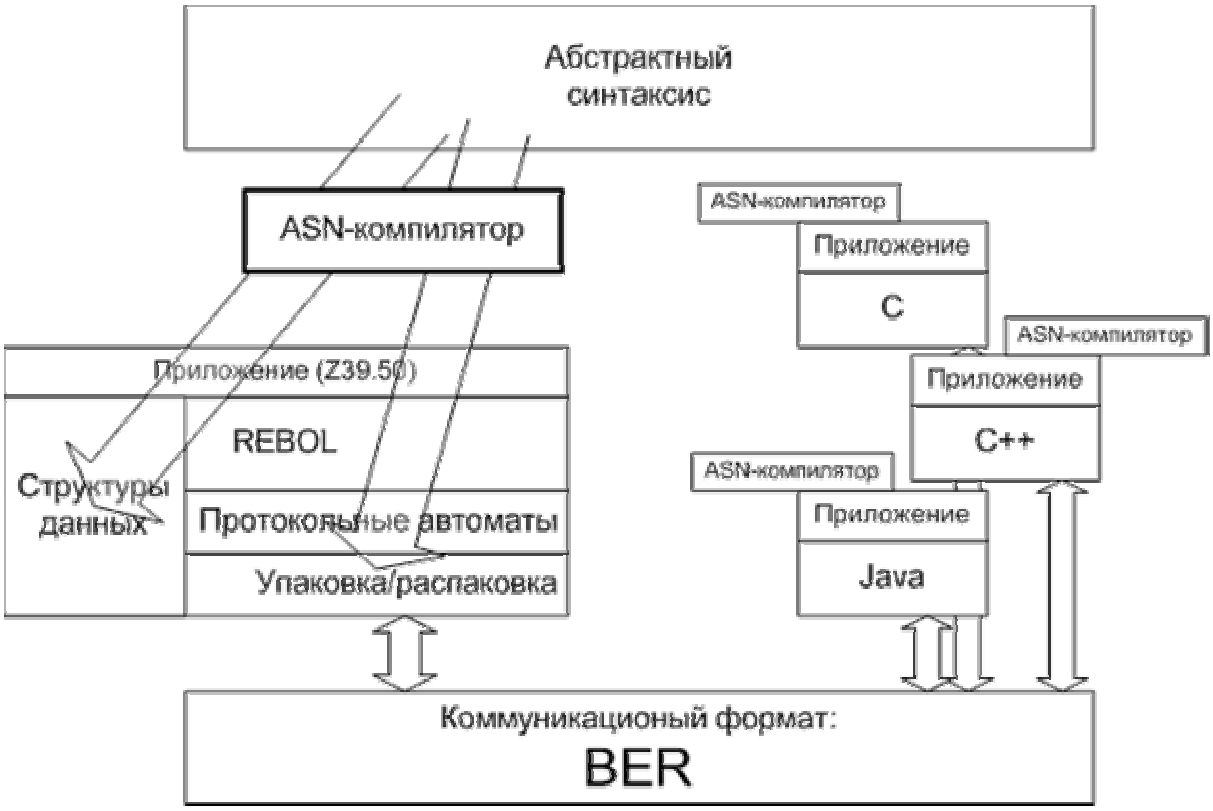

Рис.1. Функции ASN-компилятора

Потенциально структуры данных, которые можно описать с использованием ASN.1, чрезвычайно богаты и разнообразны. Однако стандарт протокола Z39.50 использует только т.н. примитивные типы данных (целые числа, логические данные, `NULL` и различные виды строк) и лишь два составных типа: `SEQUENCE` (с вариантом `SEQUENCE OF`) и `CHOICE`. Совершенно не используются, например, средства маркировки мест будущих расширений и средства параметризации.

Преобразование примитивных типов данных не представляет труда соответствующие конструкции присутствуют во всех языках программирования и синтаксически похожи. REBOL не представляет исключения. Сложности возникают при преобразовании составных типов, но здесь REBOL оказался исключительно удобным целевым языком.

Основная синтаксическая единица языка REBOL – блок. Блок REBOL представляет последовательность слов (лексических единиц REBOL) или блоков, заключенную в квадратные скобки. Это позволяет установить изоморфизм между описаниями типов ASN.1 и блоками REBOL, например:

```
      ASN:

Unit ::= SEQUENCE{
  unitSystem  [1] InternationalString OPTIONAL,   -- e.g. 'SI'
  unitType    [2] StringOrNumeric OPTIONAL,  -- e.g. 'mass'
  unit        [3] StringOrNumeric OPTIONAL,  -- e.g. 'kilograms'

  scaleFactor [4] IMPLICIT INTEGER OPTIONAL  -- e.g. 9 means
10**9

}
      REBOL:
Unit: [

  unitSystem  [1 InternationalString OPTIONAL ]   ; e.g. 'SI'

  unitType    [2 StringOrNumeric OPTIONAL ]   ; e.g. 'mass'
  unit        [3 StringOrNumeric OPTIONAL ]   ; e.g. 'kilograms'
  scaleFactor [4 IMPLICIT INTEGER OPTIONAL ]  ; e.g. 9 means
10**9
]
```

Средство реализации такого преобразования чрезвычайно просто – это регулярное выражение (используем нотацию POSIX):

```
([[:space:]]+)\(?#  Сохраним пробелы в первом найденном объекте
    - для сохранения форматирования
)[(?#
  Открывающая квадратная скобка числовой метки ASN
      )[[:space:]]?([[:digit:]]+)(?#
  Второй найденный объект - Числовая метка ASN
)[[:space:]]?\](?#
  Закрывающая квадратная скобка числовой метки ASN
)([^,]+)(?#
  Третий найденный объект -
    тип ASN и его параметры
    (IMPLICIT, OPTIONAL и т.д.)
),?(?#
  Запятая отсутствует в последней строке
    описания типа ASN
)[[:space:]]+?(?#
  Необязательный пробел перед возможным комментарием
)(--(.*))?$(?#
  Четвертый найденный объект -
    необязательный комментарий,
    простирающийся до конца строки;
    префикс комментария ASN необходимо убрать,
    поэтому текст комментария помещается
    в пятый объект
)/(?#
  Замена (комментарий в REBOL начинается с ";")
)\1[\2 \3] ;\5
```

В результате доступ к, например, названию системы единиц измерения в Z-сервере буде осуществляться с использованием переменной REBOL `Unit/unitSystem/2` (число 1, стоящее на первом месте в соответствующем блоке, является меткой коммуникационного формата и будет использовано программами упаковки/распаковки). Комментарии являются, в соответствии с ASN.1, частью стандарта протокола, и поэтому должны быть сохранены в коде реализации.

Точно так же можно поступить и с другим составным типом, присутствующим в ASN-описании Z39.50 – `CHOICE`. Однако при преобразовании этого типа можно пойти дальше.

Дело в том, что если тип `SEQUENCE` используется для передачи содержательной информации, то тип `CHOICE` используется почти исключительно для передачи информации, управляющей протокольными автоматами. REBOL же использует блоки не только для хранения сложных структур данных. Блоки REBOL могут хранить код на этом языке, который может исполняться. В результате было решено при преобразовании `CHOICE` пойти дальше и добавить в порождаемый блок еще одну строку, описывающую реальный выбор, сделанный в полученных сервером данных, например:

ASN.1:

```
PDU ::= CHOICE{
    initRequest      [20] IMPLICIT InitializeRequest,
    initResponse     [21] IMPLICIT InitializeResponse,
    searchRequest    [22] IMPLICIT SearchRequest,
    searchResponse   [23] IMPLICIT SearchResponse,
    presentRequest   [24] IMPLICIT PresentRequest,

    presentResponse  [25] IMPLICIT PresentResponse,
-- ряд строк опущен для краткости
    close            [48] IMPLICIT Close
}
```

REBOL:

```
PDU: [
    CHOICE           false
    initRequest      [20 IMPLICIT InitializeRequest]
    initResponse     [21 IMPLICIT InitializeResponse]
    searchRequest    [22 IMPLICIT SearchRequest]

    searchResponse   [23 IMPLICIT SearchResponse]
    presentRequest   [24 IMPLICIT PresentRequest]
    presentResponse  [25 IMPLICIT PresentResponse]
; ряд строк опущен для краткости
    close            [48 Close]

]
```

В переменную CHOICE процедура распаковки записывает имя того типа данных, который действительно был получен.

В результате полученный блок можно скопировать и затем заменить его внутренние блоки на блоки, описывающие действия протокольного автомата:

```
PDU-autom: [
    CHOICE           none

    initRequest      Init-recvd
    initResponse     proxy-Init-recvd
    searchRequest    search
    searchResponse   proxy-searchResponse
    presentRequest   present
    presentResponse  proxy-presentResponse
; ряд строк опущен для краткости
    close            close
]
```

Это позволяет предельно упростить сам протокольный автомат:

```
switch PDU/CHOICE PDU-autom
```

Для реального применения описанной процедуры пришлось выполнить дополнительную работу по ручному редактированию ASN-описания протокола Z39.50. Дело в том, что текст стандарта содержит несколько десятков описаний с вложенными составными типами, например (определение запроса в обратной польской записи):

```
RPNStructure ::= CHOICE{
    op    [0] Operand,

    rpnRpnOp  [1] IMPLICIT SEQUENCE{
        rpn1               RPNStructure,
        rpn2               RPNStructure,
        op                 Operator
    }

}
```

Непосредственное применение описанной процедуры к такому типу невозможно, поэтому подобные структуры разбивались введением промежуточных типов (это допускается, поскольку сохраняет семантику описания [19]):

```
RpnRpnOp SEQUENCE{
    rpn1 RPNStructure,
    rpn2 RPNStructure,
    op   Operator

}
RPNStructure ::= CHOICE{
    op    [0] Operand,
    rpnRpnOp  [1] IMPLICIT RpnRpnOp
}
```

Такие действия чреваты ошибками, поэтому их необходимо автоматизировать. Это можно сделать и с использованием регулярных выражений, однако более перспективным представляется построение "настоящего" компилятора. К счастью, REBOL включает механизм для построения синтаксически управляемых анализаторов, а соответствующая часть грамматики ASN.1 можно преобразовать в эффективно реализуемую `LL(1)` грамматику (в отличие от полной грамматики ASN.1, обладающей сильной неоднозначностью [21]). Эта работа сейчас находится в стадии выполнения и будет завершена в сентябре текущего года.

Текущее состояние разработки отражено на сайте:
`http://www.z.nw.ru`

Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант 00-07-90251

Литература

1. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50–1995).–Bethesda, MD: NISO Press, 1995.–180pp. ISBN: 1–880124–22–X.– [http://www.loc.gov/z3950/agency]
2. Z39.50 Revision (Z39.50–2001).– [http://www.loc.gov/z3950/agency/revision/revision.html]
3. Information Technology – Abstract Syntax Notation One (ASN.1): Specification of Basic Notation.–International Standard ITU–T Rec. X.680 (1997)//ISO/IEC 8824–1:1998.
4. Information Technology –ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonic Encoding Rules (CER) and Distinguished Encoding Rules (DER),.–International Standard ITU–T Rec. X.690 (1997)//ISO/IEC 8825–1:1998.
5. Margaret St. Pierre. Z39.50 for Full Text Search and Retrieval.– [ftp://ftp.loc.gov/pub/z3950/articles/margaret.pdf]

6. John Kunze. Basic Z39.50 Server Concepts and Creation.– [ftp://ftp.loc.gov/pub/z3950/articles/john.pdf]
7. Ralph LeVan. Creating a Z39.50 Client.– [ftp://ftp.loc.gov/pub/z3950/articles/ralph.pdf]
8. The Isite Information System.– [http://awcubed.awcubed.com/Isite/index.html]
9. CNIDR Isite.– [http://www.cnidr.org/software/Isite/Isite.html]
10. Kevin Gamiel, Nassib Nassar. Structural Components of the Isite Information System.– [ftp://ftp.loc.gov/pub/z3950/articles/cnidr.pdf]
11. OCLC BER Utilities.– [ftp.rsch.oclc.org/pub/BER_utilities]
12. OCLC Z39.50 Client API.– [ftp.rsch.oclc.org/pub/SiteSearch/z39.50_client_api]
13. YAZ.– [http://www.indexdata.dk/yaz/]
14. Zebra.– [http://www.indexdata.dk/zebra/]
15. ASN–компиляторы и другие программные инструменты для ASN.1.– [http://asn1.elibel.tm.fr/en/links/#tools]
16. REBOL.– [http://www.rebol.com]
17. Dubuisson O. ASN.1. Communication between Heterogeneous Systems.– OSS Nokalva, 2000.–ISBN 0–12–6333361–0.–p.463
18. Information technology – Open Systems Interconnection – Presentation service definition.–International Standard ITU–T Rec. X.216 (1994)//ISO/IEC 8822:1994.
19. Information technology – ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER). Canonic Encoding Rules (CER) and Distinguished Encoding Rules (DER).–International Standard ITU–T Rec. X.690 (1997)//ISO/IEC 8825:1998.
20. Dubuisson O. ASN.1. Communication between Heterogeneous Systems.– OSS Nokalva, 2000.–ISBN 0–12–6333361–0.–p.121
21. Ibid.–p.469