CONTIPにおけるEXFOR辞書とNRDF辞書

EXFOR Dictionaries and NRDF Dictionaries in CONTIP

札幌学院大学社会情報学部
千葉 正喜

Masaki CHIBA
Sapporo Gakuin University

Abstract

EXFOR Dictionaries and NRDF Dictionaries are installed in CONTIP, which is a integrated Nuclear Reaction Database System designed based on the IntelligentPad architecture. This report describes (1) the design of UniSQL classes for these dictionaries, (2) IntelligentPad Tools developed for loading the dictionaries and for handling entries in the dictionaries. Pad Tools developed for the CONTIP are workable on any computers connected to Internet. The CONTIP database has CGI interface and the Pad Tools of the EXFOR and NRDF Dictionaries are developed on VisualWorks platform.

1．はじめに

IntelligentPad[1,2]アーキテクチャをベースに核反応データベースサービスシステムCONTIP（Colligation of Nuclear Reaction Data and Tools with IntelligentPad）を開発している。このCONTIPにおいてはNRDF[3,4]のデータに対してもEXFOR[5]のデータに対しても一様な利用インターフェースおよび検索とその後の加工処理がシームレスに繋がる利用環境の構築を目指している。このCONTIPが管理すべきデータには、NRDFとEXFORの元々異なる二つのフォーマットのデータ、およびそれぞれの用語辞書のデータがある。

このNRDFとEXFORのそれぞれの用語辞書をCONTIPのデータベースとして利用できるようにした。そこで、ここでは、NRDFとEXFORの辞書データの構造と内容を述べて、これらのデータを格納するUniSQLのクラスを定義すると共に、データのローディングと格納したデータエントリの検索または保守のために開発したパッドツールについて述べる。最後に、今後の課題をまとめる。

2．NRDF辞書

2．1　辞書のファイルの形式と内容

オリジナルの NRDF 辞書では、用語が F、V、W、S、C と E の 6 のタイプに区分されている。このうち、C と E タイプの辞書は VOS システムに開発された NRDF システムに固有であるため、CONTIP システムでは使われない。それぞれのタイプの用語エントリレコードの集合をそのタイプの辞書とすると、このタイプと辞書の対応は次の表のようになる。

TYPE	辞書
F	項目名辞書
V	項目値辞書
W	単語辞書
S	システム辞書
C	コマンド辞書
E	コマンド使用例辞書

これら辞書タイプとその内容は次のようになっている。

F タイプ：この辞書のエントリーは、既に定義されている項目名である。

V タイプ：この辞書には、項目名に対してその値として記述できるコードが登録されている。

W タイプ：F や V の辞書の各エントリーは、一つまたはいくつかの単語コードを組み合わせて作られている。この W 辞書には、F または V のエントリー作成で用いられる単語コードが登録されている。

S タイプ：この辞書には、NRDF データの記述形式を定義しているシステム定義用語を登録している。

C タイプ：VOS のために開発された検索システム利用のためのコマンド・コードが登録されている。CONTIP では用いられない。

E タイプ：コマンド・コードの使用例が登録されている。CONTIP では用いられない。

NRDF 辞書のファイルは、各辞書タイプごとに一つの順編成ファイルになっている。ファイル上の辞書エントリーの記述形式は、各タイプとも同一と見ることができ、それは次のようになっている。

（1）各辞書ファイルはレコードの集合である。その 1 レコードは 80 バイトである。

（2）辞書エントリーの始まりは、先頭が空白でないレコードで示される。その後に続く先頭が空白のレコードはこのエントリーの記述に属するレコードである。

（3）辞書エントリの各レコードは、次の 4 種類が区別される。

1 行目：　辞書エントリーの見出し語レコードで、このエントリーで定義されるコード(CODE)が記述される。

2行目：10カラム以降にコードの展開形(EXPANSION)が記述されるレコードである。
3行目以降： 10-11カラムが「/*」ならば，12カラム以降にフリーテキストでコードの説明（EXPLANATION）が記述されるレコードである。このレコードは複数レコードになりうる。
おなじく「/+」ならば，この行の12カラム以降に、そのコードが表す内容が「属性名＝値；」の形式で記述されるレコードである。この記述は1レコード中に複数回繰り返す。
（1）属性名とその取り得る値は次のとおりである。下の表で記号 | は‘または’を表す。

属性名	内容
TYPE	F | V | S | C | E | W
CLASS	1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 2桁までの数字，一桁の数字の場合，先頭は空白，複数ある 時は「,」で区切ってならべる
RATE	単位の変換係数で、数値の表現形式はE12.5になっている
BASE	基底の単位コードを値にとり、CLASS=14の場合に書かれる
SOURCE	EXFOR | NRDF
DATE	YY-MM-DDの形式で書かれる日付
COUNTRY	4桁の国コード
FLAG	使わなくなったコードを0で示す

2．2　NRDF辞書で識別される属性

NRDF の各辞書で共通に識別される情報項目は、（1）辞書エントリーの見出し語（CODE）、（2）CODE の展開形（EXPANSION）、（3）CODE の説明テキスト（EXPLANATION），（4）CODE が属する辞書のタイプ（TYPE），（5）辞書エントリー更新日付（UDATE），（6）CODE が廃止になっていることを示すフラグ（FLAG）である。
V タイプ辞書にのみで識別される情報項目は、（7）CODE がどこで使われるかを示す（CLASS），（8）CODE が物理量の単位である場合、BASE の単位との換算率（RATE），（9）CODE が単位の場合、その基底単位（BASE）、（10）CODE が NRDF または EXFOR のどちらの由来かを示す（SOURCE），（11）CODE が示す内容がどの国に属するかを示す（COUNTRY）である。

3．NRDF辞書クラスの定義

NRDF の各辞書はこれらの識別される属性の部分集合で表すことができる。これら辞書のデータを UniSQL で管理するため，共通する属性は NrdfDiction クラスで定義し，それぞれのタイプの辞書はこのサブクラスとして定義している。

（1）各辞書のスーパークラスの定義

```
NrdfDiction class
(
 code      character varying not null,
 expansion          character varying,
 explanation        sequence(character varying),
 dictType    character(1)   not null,
 udate      date,
 flag    character(1)
 );
```

（2）Fタイプ辞書の定義：

```
NrdfDictionF as subclass of NrdfDiction
(
 dictType    shared 'F'
);
```

（3）Sタイプ辞書の定義：

```
NrdfDictionS   as subclass of NrdfDiction
(
 dictType    shared 'S'
);
```

（4）Vタイプ辞書の定義：

```
NrdfDictionV   as subclass of NrdfDiction
(
 dictType    shared 'V',
 codeclass          set(smallint),
 rate      character(12),
 base character(40),
 source   character(5),
 country             character(7)
);
```

（注）RATE　の内容は，{- ，　} 9.99999E{+,-}99 の形式の文字列である．

（5）W タイプ辞書の定義：

```
NrdfDictionW as subclass of NrdfDiction
(
 dictType    shared 'W'
);
```

4．EXFOR 辞書データ

4．1 Transmission File

EXFOR 辞書は、国際原子力機関核データ部（NDS）によってその内容が更新されて、定期的に EXFOR Dictionary Transmission File に編集されて各データセンターに配信される。この EXFOR Dictionary Transmission File が CONTIP における EXFOR 辞書システムの入力データである。

この EXFOR Dictionary Transmission File は NDSOPEN から FTP で配信しているので、次のようにして入手する。ここで、trans.9073 は、Transmission File のファイル名である。EXFOR Dictionary Transmission File の場合は、trans. に続く番号は 9 で始まる。この 9 を除いた数値がファイル作成順番になっている。

```
ftp iaeand.iaea.or.at
name: NDSOPEN
cd trans
get trans.9073
```

EXFOR Dictionary Transmission File は、EXFOR システムで使われるすべての辞書を含み、次のシステム識別子レコードを用いて編成されている。EXFOR Dictionary Transmission File の編成で用いられるシステム識別子レコードは、TRANS, ENDTRANS, DICTION と ENDDICTION の 4 種類である。これらのシステム識別子レコードは 80 バイトで次の一般的な形式を持つ。

1-11 カラム：システム識別子欄である。ここに TRANS, ENDTRANS, DICTION または ENDDICTION のいずれかのシステム識別子が左詰めで置かれる。
12-22 カラム：N1 欄で、ここには整数が右詰めで記述される。その意味は、システム識別子による。

23-33 カラム：N2 欄で、整数が右詰で記述される。その意味は、システム識別子による。
34-66 カラム：フリーテキスト欄である。
67-79 カラム：レコード識別子欄である。
80 カラム：フラグ欄である。

EXFOR Dictionary Transmission File 編成の形式は次のようになっている。
第 1 レコードは、TRANS レコードである。N1 欄はファイル交換識別子で、先頭のファイル識別子文字は 9 である。それに続く 3 桁はファイル作成順番である。N2 欄は、6 桁のファイル作成日付で YYMMDD の形式である。レコード識別子は、30000 である。
最後のレコードは ENDTRANS レコードで、この N1 は空白、N2 はこのファイルに含まれる辞書の数である。レコード識別子は、そのすべてのカラムが“9”である。
すべての EXFOR 辞書はこれら TRANS と ENDTRANS レコードの間に辞書識別番号順に置かれる。辞書識別番号は 001 から 099 の範囲の値をとる。
各辞書は、その先頭と最後がそれぞれ二つのシステム識別子レコード DICTION と ENDDICTION で区切られている。DICTION は各辞書の最初のレコードで、N1 と N2 の内容は次のようになっている。

N1：辞書識別番号
N2：最後に更新した日付（YYMMDD）
34-66 カラム：辞書の内容を記述するフリーテキスト
レコード識別子：67-71 が“30000”、72-74 が辞書識別番号、75-79 がレコードシーケンス番号で“00001”

ENDDICTION は各辞書の最後のレコードで、この場合 N1 と N2 の内容はそれぞれ次のようになっている。

N1：DICTION と ENDDICTION を除く、辞書内のレコード数
N2：未使用で空白
レコード識別子：レコードシーケンス番号が“99999”の他は DICTION レコードと同じ

4．2 エントリーの内容と構造

辞書の各エントリーは、キーワードまたはコードとその説明、およびフラグから構成されている。これらの情報は、66 カラムから 79 カラムにレコード識別子欄を持つ 80 バイトレコードのそれぞれのフィールドに記述されている。
この辞書レコードには、次のように欄が設定されている。
（1）キーワードまたはコード欄： EXFOR で用いられるキーワードまたはコードを記述

する欄である。この欄は通常レコードの 1-11 カラムに置かれ、キーワードまたはコードはこの欄に左詰めになっている。この欄がこれより長いものもある。しかし、キーワードが 10 字を超えることはない。コードの辞書の場合、3 字または 5 字に限定されているものもある。

（2）説明欄：この欄は通常 12 カラムから始まり 66 カラムで終わるが、例外もある。

（2）フラグ欄：80 カラムはフラグが置かれ、このレコードにあるコードが次の条件にあることを示す。O フラグの場合：新しくデータを送るときにはそのキーワードまたはコードは使えないが、以前に送られたデータ中にはまだこのコードは存在していることを示す。コードを使わなくなった理由とそれに代わるコードがあればそのコードをフリーテキストで説明されている。X フラグの場合：廃止になった施設、雑誌または定期報告書を指すコードであることを示す。

各辞書にはキーワードまたはコードとその説明のための情報項目が含まれる。これらの情報はそれぞれ上で述べてキーワードまたはコード欄と説明欄に次のように記述されている。

キーワードまたはコード欄：この欄がキーワードであるのは次の 4 つの辞書である。

システム識別子辞書： 辞書番号 1
情報識別子辞書： 辞書番号 2
データヘッディング辞書： 辞書番号 24
データ単位辞書： 辞書番号 25

辞書 24 と 25 に定義されているデータヘッディングとデータ単位は COMMON セクションと DATA セクションで使われる。その他の辞書では、この欄にコードが定義されている。これらの辞書に定義されるコードは BIB セクションにおける特定の情報識別子キーワードのもとで使われる。

説明欄：この欄には説明が記述されるが、その説明は
（1） フリーテキスト
（2） 展開形
（3） 展開形とそれに続くフリーテキスト
のいずれかで与えられる。
展開形は、特定の辞書に用意されていて、コード自身の説明またはコード自身を思い出しやすいような表現になっている。展開形はカッコで括られている。すなわち開きカッコは説明欄の第 1 カラム(通常はカラム 12)にある。辞書の 1 エントリに対してただ 1 組のカッコが関連させられる。展開形は、一般には 1 レコードの説明欄の長さに限定されるが、展

開形が以下のレコードの説明欄に続く辞書もある。

フリーテキストは展開形の閉じカッコの直後に続くか、または展開形が無い場合は説明欄の最初のカラムから始まる。フリーテキストは説明欄の範囲で任意のレコードに続くことがある。フリーテキストにカッコが含まれることがあるが、フリーテキストの部分である左カッコは説明欄の最初のカラムにあることはない（それは展開形があることの印になる）。

幾つかの辞書では、この欄にその他のコード情報が含まれている。詳細はそれぞれの辞書の項で述べる。

EXFOR 辞書一覧表

辞書番号	辞書名	コード長	展開形の有無
1	System Identifier	≦10	-
2	Information Identifiers	≦10	あり
3	Institutes	5 to 7	あり
4	Reference Type	1	あり
5	Journals	≦6	あり
6	Reports	≦11	-
7	Conference and Books	≦10	あり
8	Element	≦6	あり
9	Chemical Compounds	7 to 10	あり
13	Particle	≦3	あり
15	History	1	あり
16	Status	≦5	あり
17	Rel-Ref	1	あり
18	Facility	≦5	あり
19	Incident Source	≦5	あり
20	Aditional Results	≦5	あり
21	Method	≦5	あり
22	Detectors	≦5	あり
23	Analysis	≦5	あり
24	Data Headings	≦10	-
25	Data Units	≦10	-
27	Nuclides	≦10	-
28	Incident Particles	≦3	あり
29	Product Particles	≦3	あり
30	Process	≦3	-

31	Branch	≦5	-
32	Parameter	≦3	-
33	Particles Considered	≦3	あり
34	Modifiers	≦3	-
35	Data-Types	≦5	あり
36	Quantities	≦44	あり
37	Result	≦5	あり
42	Cinda Quantities	≦3	あり
43	NLIB for evaluated lib.	≦2	-
50	List of Dictionaries	≦2	-

4．3 個別辞書について附加情報

辞書2．Information Identifiers

各キーワードに対する最初のレコードは、次の形式になっている。

カラム 1-11：キーワード

カラム 12-33：展開形

カラム 34-44：このキーワードが必須かどうかを示すコード

REQ	必要
XREQ	関係が無い場合を除いて必要
AREQ	一連のキーワードの一つが必要
BREQ	一連のキーワードのうち関係があるもの、ただし少なくとも一つが必要
OBS	そのキーワードはもう使わない、ただし古いエントリには有り得る

カラム 45-55：コード情報の定義

RCODE	コードが必要
OCODE	任意
OCODE+	任意、もし与えたなら、フリーテキストでも繰り返す

カラム 56-66：使われる辞書へのポインターで、‘+’はコード情報があることを示す

フリーテキストの情報は、以下に続くレコードのカラム 12-66 に含まれる。

辞書 3. Institute

7文字のコード ABBBCCC は、次のように構成されている。

A　サービスエリアコードで、1，2，3、または4である。

BBB　国別コード

CCC　実験施設コード（3文字より少ないことがあり、その場合は左寄せ）

3文字の実験施設コードは、EXFORで使われるすべての実験施設、大学、研究所、機関、委員会のコードで、一意である。コードが国のみを区別しているときは、国別コード欄の情報が実験施設コード欄で繰り返されている。したがって、国別コードと同じ実験施設コードは無い。旧コードに対してはO、廃止されたコードに対しては80カラムにXと印がつけられている。置き換えのコードがある場合は、与えられている。この辞書はコードでソートされていて、エリアと国でグループ化されている。

辞書4. Reference Type

説明欄のカラム56-61が“DICTn”にあらかじめ取られている以外は標準形式である。“DICTn”は 辞書番号が“n”の辞書を指し、その辞書にはその参照型で使われる参照コードを含む。カラム56-61が空白のときは適用する辞書がないことを示している。

辞書5．Journals.

実際の雑誌コードは 4 文字以下に限定されている。雑誌が幾つかの部分に別れているときは、その部分も雑誌コードと共に辞書に含まれていて、それがスラッシュで区切られている。したがって、完全なコードは6文字以下である。カラム63-66には（出版している国の）エリアコードと国別コードがある。

展開形は、雑誌の標題に一般的に採用されている様式に従っている。しかし、ある種の省略形は明確化のため展開されている。

旧コードはOと廃止コードにはXと80カラムに印をしてある。この辞書はコードでソートされている。

辞書6．Reports.

この辞書の各コードは、実際の報告書番号の前にくる文字列でできている。辞書にあるコードの最後の文字は通常は － であるが、報告書コードが11文字で12番目の文字が － であるときは例外である。そのような場合は、辞書から － が落ちている。

報告書番号が与えられていない年次報告書に対しては、EXFORは次のように報告書コードを割り当てる： コード A-、続いて 3 桁の実験施設コード。コード化するときは、このコードにレポートを出した年を続ける。例えば、 A-ARK-84

カラム60-66には報告書を出している研究所の7桁のInstitute Codeがある。

旧コードと廃止コードに対しては、それぞれ80カラムにOまたはXと印がつけられている。辞書は、研究所コード、研究所コード内では報告書コードでソートされている。

辞書7. Conferences and Books

Conference コードは、コードの最初の 2 桁でその会議が行われた年を示し、その後が会議の行われた場所になっている。
Book コードは、本の短い標題または第一著者の姓になっている。

辞書 9. Chemical Compounds

一般的な化合物コード‘CMP’は、この辞書に登録することなく、任意の原子に一般形（Z-S-CMP）の形式で結合できるので、それが辞書に登録されているのは特別な場合である。

辞書 16. Status

この辞書は、説明欄の 66 カラムがフラグのためにとってある。それ以外は標準形式である。このフラグは、コードにアクセッション番号欄が続くことを示している。

S　　コードにアクセッション番号欄が続き得ることを示す
R　　コードには常にアクセッション番号欄が続くことを示す。

辞書 24. Data Headings

データ標題は COMMON セクションと DATA セクションでデータ欄の内容を定義するために使われる。コードの展開形は無い。辞書 24 と辞書 25 を通して、コードは一意である。すなわち、データ標題と同じデータ単位は無い。旧コードはカラム 80 に O の印がある。
カラム 66 はフラグとして予約されている。このフラグはチェック目的に使われ、カテゴリーと各カテゴリー内の同族（または独立変数の型）を定義していて、次の表の枠組みになっている。

族	フラグ 変数	フラグ 関連物理量	カテゴリ
Incident energy	A	B	1
Resonance energy	C	D	1
Secondary energy	E	F	1
Angle of outgoing particle	G	H	1
Product charge	I	*	1
Product mass	J	*	1
Secondary Liner momentum	L	*	1

Linear momentum	M	*	1
Coefficient number	N	*	1
Neutrons out	O	*	1
Protons out	P	*	1
Secondary effective mass	S	*	1
THIKNESS	K	*	2
FLAG	Z	*	2
TEMP	8	9	2
HL	6	7	2
J	4	*	2
Parity	0	*	2

* はヘッディングが存在しないことを示す。
カテゴリー1は独立変数に属する。
カテゴリー2は附加情報に属するが、ある場合には独立変数として機能することもありうる。

辞書 25. Data Units

データ単位は、データ標題の下で各欄の内容の単位を定義するために COMMON セクションと DATA セクションで使われる。コードは辞書 24 と辞書 25 内で一意である。すなわち、データ単位と同じデータ標題はない。
この辞書の形式は次の通りである。

カラム 1-10　コード
11　空白
12-14　コードの説明(展開形はない)
45-48　次元コード
49-55　空白
56-66　変換係数

次元コードは、辞書 36 との間でお互いにリンクしている。辞書 36 は物理量の辞書で、ここでも次元コードが与えられている。これで表の物理量と単位に矛盾が無いことを計算機でチェックできるようにしている。

変換係数は浮動小数点数で、同じ次元の単位を標準の単位に変換するのに使える。標準の単位の例：

エネルギー　エレクトロンボルト

角度	度
時間	秒
長さ	メートル
断面積	バーン

辞書 27. Nuclides

この辞書の形式：

カラム 1-11　核子コード

12-26　カッコで囲まれたフラグである。フラグの意味はその位置により定義されている。

フリーテキストがあるときは、これに続くレコードがありカラム 12 から始まる。

核子コードは Z-S-A の形式である。ここで、

Z　3 桁までの電荷数、先頭に 0 はつかない

S　元素記号、1 文字または 2 文字

A　3 桁までの質量数、先頭の 0 はつかない、0 は自然同位元素複合物を表す

12-26 カラムは次の構造になっている。

カラム 12　'('

カラム 12-23　各カラムはフラグまたは空白である。

カラム 13　REACTION SF1（SF2=0）のために用いられる

'1' は有効であることを示す

'X' は通常ではないことの警告である

カラム 14　REACTION SF2 のために用いられる

'2' は有効であることを示す

カラム 15　REACTION SF3、REACTION SF4、REACTION SF7、HALF-LIFE、DECAY-DATA、DECAY-MON、RAD-DET、PART-DET と EN−SEC のために用いられる。

'3' は有効であることを示す

'Z' は REACTION SF3、PART-DET と EN−SEC を除いて有効であることを示す。このような場合は核子コードではなく対応する粒子コードが使われる

カラム 16　REACTION SF1（SF2=0）で用いる

'4' は有効であることを示す

カラム 17　Fission Product で用いられる

'F' は友好であることを示す

カラム 17-21　現時点まで使われていない

カラム 22　　CINDA で用いられる
　　'C' は有効であることを示す
　　'T' は理論の仕事でのみ有効であることを示す
カラム 23　　安定核であることを示すために用いられる
　　'S' は安定であることを示す
カラム 24-25　isomer 欄である
　　空白ならば核子が isomer state を持たない
　　または、数ならば metastable state の最大値を示す
カラム 26　　')'

辞書 34.　Modifiers (REACTION)

この辞書には、REACTION コードで用いる修飾子の展開形が与えられているが、この修飾子コードは、辞書 36 には含まれていない。

辞書 36.　物理量 (REACTION)

この辞書の形式は次のようになっている。

カラム 1-18　　物理量コード、これが 18 文字を越える場合、このレコードのその後にに続き、66 カラムに '9' が書かれている。次元コード、展開形は次のレコード以降の割り当てられたカラムに書かれる。したがって、物理量コードの最大長は 44 である。

カラム 19-21　次元コード、辞書 25 へのリンクになっており、物理量と単位に整合性のチェックに使える。

カラム 22　　フラグ、レゾナンスパラメータの場合に フラグとして '.' が書かれている

カラム 23-66　展開形とフリーテキスト

辞書 42.　CINDA[1] Quantities

この辞書のコード欄(1-10 カラム)には次のように記述されている。

1-3 カラム：　3 文字のコード、左詰

4-8 カラム：数字コード、整数が右詰

辞書 43.　NLIB for evaluated libraries

この辞書のコード欄(1-10 カラム)はライブラリーの数値コードを定義している。

1-2 カラム：　数字コード、整数が右詰

---

[1] A specialized bibliography and index on neutron nuclear data operated jointly by NNDC,NEA-DB,NDS and CJD

辞書 50.　List of dictionaries

このコード欄(1-10 カラム)は辞書番号である。

1-2 カラム：　数字コード、整数が右詰

5．EXFOR 辞書データのデータベーススキーマ

CONTIP では、EXFOR 辞書データは UniSQL/X データベース管理システムを介して利用する。上で述べた各 EXFOR 辞書を UniSQL に格納するためのクラスを次のように設計定義する。

（1）EXFOR の各サブ辞書に対応して、一つのクラスを定義する。そのクラスの名前は、ExforDict<辞書番号>とする。これらのクラスは ExforDict クラスのサブクラスとして定義される。

（2）各サブ辞書クラスのスーパークラスとして ExforDict クラスをを定義する。このクラスには、各 EXFOR 辞書が共通に持つ属性に関する情報を保持させる。

（3）ExforDict クラスでは、次の属性を置く。

dicNo：　辞書番号で各 DICTION レコードの N1 欄の内容である。

dicName：辞書名で、各 DICTION レコードの 34-66 カラムの内容である。

uDate：　辞書の更新日付で、各 DICTION レコードの N2 欄の内容である。

explanation：各辞書について、その辞書の先頭にその全体の説明やコメントが記述されている場合に、それを内容とする。例：辞書 2、辞書 5 など。この属性を SEQUENCE VARCHAR(55)と定義して、複数行の情報を収容する。

flag：　各辞書レコードの 80 カラムの内容である。

expansion：　属性の定義をサブクラスに継承するのみである。

（4）ExforDict<辞書番号>クラスのすべてで、次の属性が定義される。

DicNo：SHARED <辞書番号> と定義している。

code：この属性はそれぞれの辞書で個別に定義している。辞書レコードのキーワードまたはコードがその内容である。辞書 42 と 43 では、データ型を SMALLINT、その他の辞書では VARCHAR でデータ長はそれぞれのコードの長さになっている。

entrySeq：辞書エントリーの順番を保存するための属性で、1 から始まる整数を値とする。

expansion：コードの展開形を‘(’ と‘)’を含めてその内容とする。属性の定義は、ExforDict クラスから継承するか、または再定義される。

expansion：キーワードまたはコードの説明の記述情報を内容とする。属性の定義は、ExforDict クラスから継承する。

（5）特定のクラスでは、さらに次のような属性が定義される。、

ExforDict 2 クラス

keyReq：このキーワードが要求されるかどうかを示すコードで、第1レコードの34-44カラムの内容である。

‘REQ’ － 必要

‘XREQ’－ 関係ない場合を除いて必要

‘AREQ’－ これらのコードの一つが必要

‘BREQ’－ 関係がある場合はこれらの各キーワードは必須であるが、少なくともその一つがなくてはならない

‘OBS’ － キーワードは廃止になっている；古いエントリーにはあるかもしれない

codeReq：コード化情報の定義で、第1レコードの45-55カラムの内容である。

‘RCODE’ － コードを必要とする

‘OCODE’ － コードは任意である

‘OCODE+’－ 任意でるが、もしあるとフリーテキストでも記述する

relDic：使われる辞書へのポインターで、第1レコードの 56-66 カラムの内容である。ここで、‘＋’はコード情報が付加されることを示す

ExforDict4クラス

RelDic：レファレンスの型に対応して使われるリファレンスコードを定義している辞書へのポインターで、この内容は‘DICT n’である。

ExforDict5クラス

country：このコードの雑誌が発行されたエリヤと国のコードがその内容である。

ExforDict6クラス

institute：このコードの報告書を発行した研究所コード(辞書3にあるコード)を内容とする。

ExforDict8クラス

ｚNo：コードの一部である元素番号をその内容とする。データ型は SMALLINT である。

elmSymbol：コードの一部である元素記号を内容とする。

ExforDict9クラス

zNo：コードの一部である元素番号を内容とし、データ型はSMALLINTである。

elmSymbol：コードの一部である元素記号が内容である

compound：コードの一部である化合物コードの部分のみがないようである。

ExforDict16 と 34 クラス

relInfo：エントリーのサブセットにコメントが附されている。そのコメントへのポインタを内容とする。コメントが附されてない場合は、空である。コメント自身はクラス Related に格納される。

ExforDict18、 19、 21、 22、 23、 24、25 と 36 クラス

category：これらの辞書のコードエントリがカテゴリ標題のもとでサブセットにまとめられている。このカテゴリ標題を内容とする

ExforDict24 クラス

checkFlag：66 カラムのチェックフラグをその内容とする。

ExforDict25

dimension：45-48 カラムにあった次元コードがこの内容である。

convFact：56-66 カラムの変換係数がその内容である。データ型は VARCHAR(11) である。

ExforDict27 クラス

zNo：コードの一部の原子番号で、データ型 SMALLINT である。

elmSymbol：コードの一部の元素記号である。

aNo：コードの一部の質量数で、データ型は SMALLINT である。

ExforDict36 クラス

dimension：19-21 カラムにあった次元コードである。

resonance：レゾナンスパラメータの場合、‘.’がその内容になっている。22 カラムに定義されていたものである。

ExforDict42 クラス

numCode：CINDA Quantity の数字コードである。データ型は SMALLINT になっている。

（6）ExforDict クラスのサブクラスとして Related クラスを定義する。このクラスは、特定の辞書(辞書 16 と辞書 34)のいくつかのエントリーに関連付けられているコメントを保持する。

relInfo：explanation のテキスト情報は、dicNo の内容と relInfo の内容が一致す

るエントリーに対するコメントである。

なお、EXFOR 辞書に対する UniSQL のクラス定義は付録１のとおりである。

６．NRDF 辞書のための PAD ツール

NRDF 辞書を取り扱うために、二つのパッドを用意している。一つはファイルの NRDF 辞書をデータベースのクラスに入力するためのパッド（NrdfDictLoadPad）、他の一つはデータベースの辞書を検索または更新するためのパッド（NrdfDictToolPad）である。

図１は、NrdfDictLoadPad に文字列入力パッド、ボタンパッド、ストリングホルダーの各パッドを接続して、NRDF 辞書をデータベースにロードできるように構成した合成パッドである。

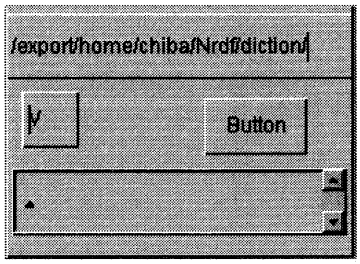

図 1 NrdfDictLoadPad

図 2 は、NrdfDictLoadPad を用いて、NRDF 辞書データベースの保守目的で利用できるように構成した合成パッドである。エントリーの検索、更新、挿入、消去のそれぞれの手続きを起動するボタンパッドを配置している。

図３は、NrdfDictLoadPad を検索利用目的のために用いるように構成した合成パッドである。#code と#dictType スロットには、PromptPadWithSend パッドが、#codeClass スロットには、NumberPadWithSend パッドが接続されている。

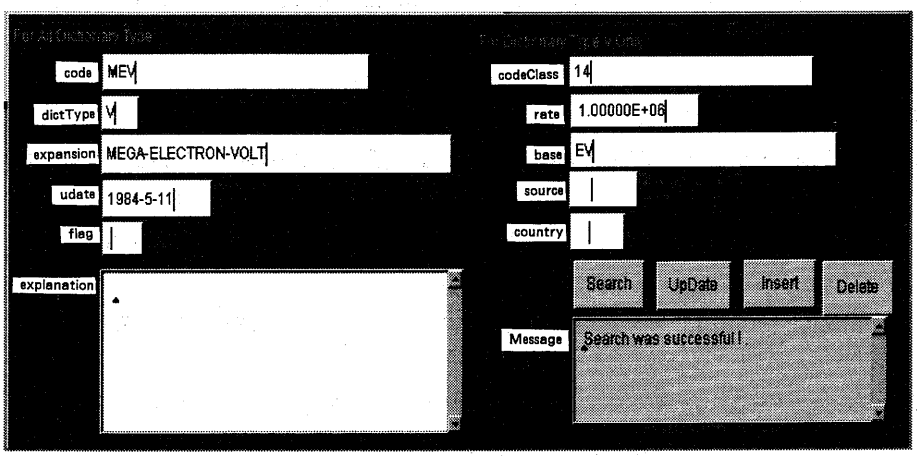

図 2 NrdfDictToolsPad

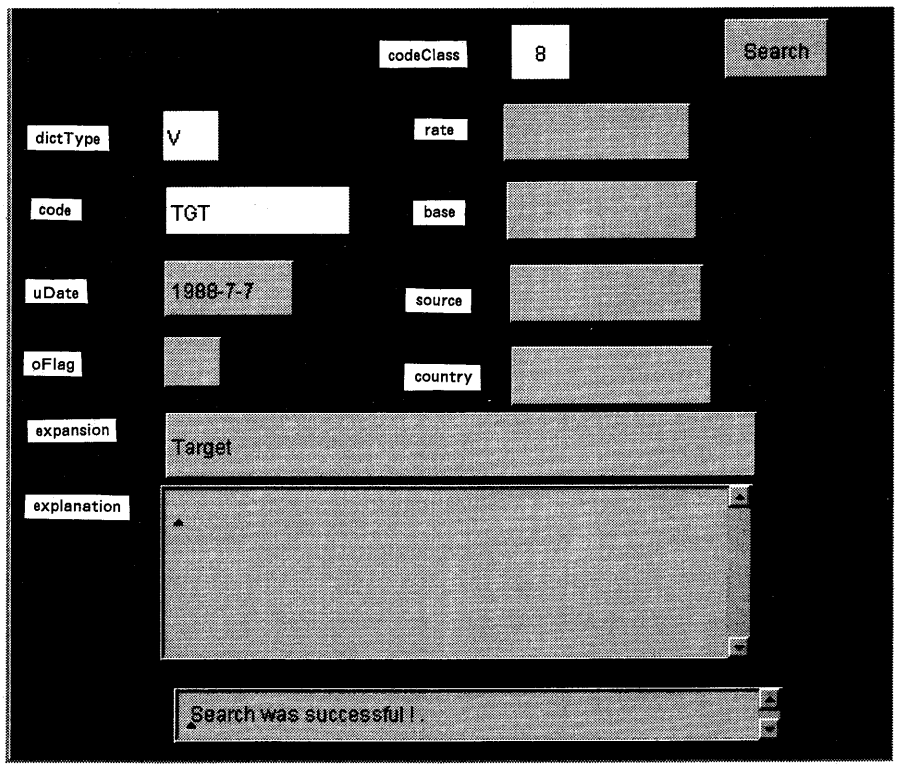

図 3 NrdfDictToolsPad customized for search only

以下で、図1の台紙になっている NrdfDictLoadPad、および図 2 と図 3 の台紙になっている NrdfDictToosPad について詳述する。

（1） NrdfDictLoadPad

この Pad は、モデル部が NrdfDictPadM、ビュー部が PadView で作られるパッドである。モデル部の NrdfDictLoadM に用意されスロットと役割は次のとおりである。

#host：データベースサーバーのアドレスを保持しているスロットで、初期値は‘nrdf.meme.hokudai.ac.jp’に設定してある。

#dbName：データベース名を保持しているスロットで、初期値を‘NRDBs’に設定してある。

#dicName：NRDF 辞書のタイプ｛W|F|V|S｝を指定する。このスロットには文字列が入力できるパッド（PromptPad など）を接続する。

#DicDirectory：NRDF 辞書ファイル｛W|F|V|S｝のディレクトリパスを指定する。このスロットには文字列が入力できるパッド（PromptPad など）を接続する。

#messageLog：辞書エントリーをデータベースのクラスに入力処理中に、このパッドが出力するメッセージを保持するスロットである。このスロットには、文字列が出力できるパッド（StringHolderPV/PadModel）を接続する。

#errCount：辞書のローディング処理中にエラーを検出した場合、検出したエラーの個数を数える。

#startColmn：入力レコードにおけるテキスト開始位置を保持する。3が設定してある。

#startButton：処理開始のトリガーを入力する。このスロットには、ボタンパッドを接続する。

（2） NrdfDictToolPad

これは、UniSQL に作られた NRDF 辞書のエントリを検索または更新する機能を持つパッドで、モデル部が NrdfDictToolsPadM、ビュー部が NrdfDictToolsPadV で定義される。モデル部 NrdfDictToolsPadM に用意されるスロットとその機能は次のとおりである。

#host：データベースサーバーのアドレスを保持している。

#dbName：データベース名を保持している。

#search：検索処理の開始トリガーを入力する。

#insert：挿入処理のトリガーを入力する。

#delete：消去処理のトリガーを入力する

#update：更新処理のトリガーを入力する。

#dictType：辞書のタイプ｛W|F|V|S｝をその値とする。ニックネームが‘dictType’であるパッド（PromptPadWithSendView/PadModel）を接続する。このスロットの値がセットされると、そのタイプの辞書のコードのリストを#codeList スロットに設定する。

#e_message：検索、挿入、消去、または更新処理でエラーを検出した場合のメッセージが設定される。
#code：辞書の見出し語の‘コード’を保持する。ニックネームが‘code’であるパッド（PromptPadWithSendView/PadModel）を接続する。
#codeList：メニューで表示する‘コードのリスト’を保持する。
#expansion：コードの‘展開形’を保持する。
#explanation：コードの‘説明テキスト’を保持する。
#udate：コードの‘更新日付’を保持する。
#flag：コードの‘フラグ’を保持する。
#rate：RATE 属性の値である。
#cdClass：コードの‘クラス’を保持する。ニックネームが‘cdClass’である（NumberPadWithSendView/NumberPadModel）を接続する。このスロットの値がセットされると、コードクラスがその値であるコードのリストが#codeList スロットに設定される。
#base：BASE 属性の値である。
#source：SOURCE 属性の値である。
#country：COUTRY 属性の値である。

ビュー部 NrdfDictToolsPadV は、sendEvent:from:with:メッセージに反応し、次のように選択メニュを表示する。ニックネームが‘code’である（PromptPadWithSendView/PadModel）からのセレクトボタンが押されたイベントが送られた場合は、#codeList が保持するリストをメニュ表示し、選択された項目を#code スロットに設定する。ニックネームが‘cdClass’である（NumberPadWithSendView/NumberPadModel）からのセレクトボタンが押されたイベントが送られた場合は、クラスカテゴリをメニュ表示し、選択されたクラス番号を#cdClass スロットに設定する。ニックネームが‘dictType’である（PromptPadWithSendView/PadModel）からのセレクトボタンが押されたイベントが送られた場合は、NRDF 辞書タイプ{F V W S}をメニュ表示し、選択された項目を#dictType スロットに設定する。

7．EXFOR 辞書のための PAD ツール

EXFOR 辞書を取り扱うために、二つのパッドを用意している。一つはファイルの EXFOR 辞書をデータベースのクラスに入力するためのパッド（ExforDictLoadPad）である。われわれが、EXFOR 辞書のエントリを修正更新することは無いと考えられるので、他の一つは検索機能のみ持つパッド（ExforfDictSearchPad）である。

図４は、ExforDictPad を用いて、EXFOR 辞書をデータベースにロードするツールを構成した合成パッドを示す。二つの NumberPad と一つの PromptPad が ExforDictLoadPad

に貼られている。左側の NumberPad にはロードすべき辞書番号を入力する。0 を入力した場合は、すべての辞書をロードする。右側の NumberPad にはロード処理した辞書番号が表示される。

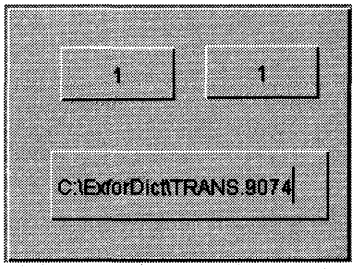

図 4　ExforDictLoadPad

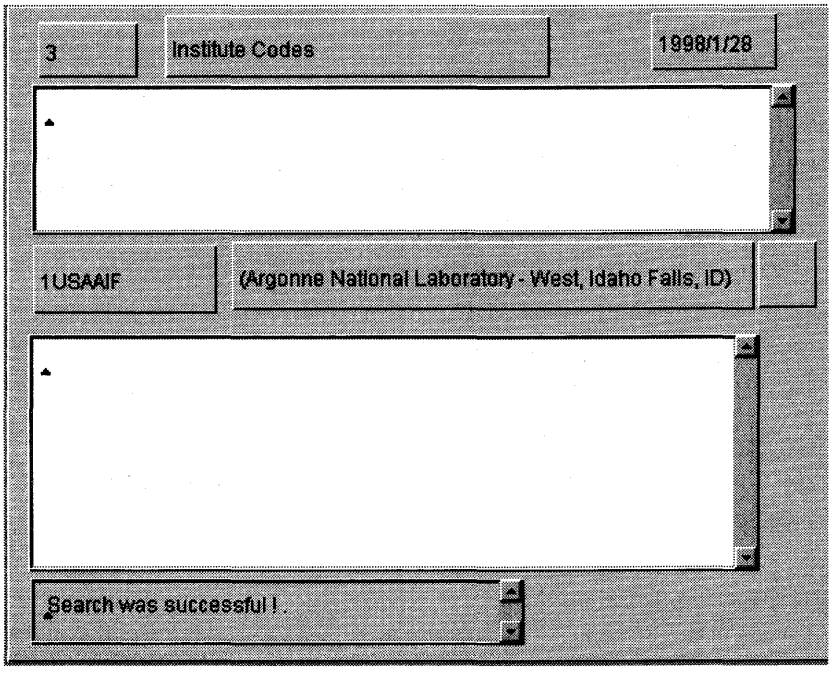

図 5　ExforDictSearchPad

図５は、ExforDictSearchPad を用いて、EXFOR 辞書データベースの検索ツールを構成した合成パッドである。この合成パッドにおいて、PromptPadWithSend パッドとNumberPadWithSend パッドを用いている。これにより、辞書番号の入力(‘３’を表示しているパッド)とコードまたはキーワードの入力（‘1USAAIF’を表示）では、このパッドをセレクトボタンでクリックするとメニューから指定できるようになっている。

以下で、図４と図５の合成パッドで台紙になっている ExforDictLoadPad と ExforDictSearchPad の構成について詳述する。

（１） ExforDictLoadPad

このパッドは、モデル部がExforDictLoadPadMとビュー部がPadViewで構成され、EXFOR Trans ファイルのすべてのEXFORサブ辞書または指定された辞書番号のサブ辞書をUniSQLデータベース（NRDBs）の各クラスにロードする。この機能を実現するため、このパッドのモデル部ExforDictLoadPadMに用意されているスロットは次のとおりである。

#host：データベースサーバーのアドレスを保持している。

#dbName：データベース名を保持している。

#transFile：EXFOR Transファイルが置かれているディレクトリを含むファイル名である。

#dicNo：ローディングすべき辞書の辞書番号である。すべての辞書をローディングするときは、数 0 を指定する。このスロットに値が設定されると処理メソッドが起動される。

（２） ExforfDictSearchPad

このパッドはEXFORのサブ辞書の一つを選択して、選択したサブ辞書のコードまたはキーワードで検索する。(ただし、辞書43と50は、コードのデータ型が整数なので、このパッドでは扱えない。）表示する属性は、コードの展開形、フラグ、コードの説明テキストになっている。サブ辞書の指定とコードの指定はメニューからも行える。このパッドはモデル部がExforDictSearchPadMとビュー部がExforDictSearchPadVで構成される。

モデル部 ExforDictSearchPadM に定義されるスロットとその機能は次のとおりである。

#host：データベースサーバーのアドレスを保持している。

#dbName：データベース名を保持している。

#dicNo：辞書番号を保持する。このスロットに値が設定されると、データベースを検索してこの辞書のキーワードまたはコードのリストを#codeList に設定する。このスロットには、ニックネームが‘dicNo’であるパッド（NumberPadWithSendView/NumberPadModel）を接続すると、辞書番号の指定がメニューから行える。

#code：コードまたはキーワードを保持する。このスロットに値が設定されると、その辞

書のエントリーを検索する。このスロットにはニックネームが‘code’であるパッド（PromptPadWithSendView/PadModeld）を接続すると、コードまたはキーワードの指定がメニューから行える。

#flag：検索されたエントリーのFLAG属性の値を保持する。

#expansion：検索されたエントリーのEXPANSION属性の値を保持する。すなわち、コードの展開形である。

#explanation：検索されたエントリーのEXPLANATION属性の値を保持する。このスロットは複数行のテキストを保持するので、それが表示できるパッド（StringHolderPV/PadModel）を接続する。

#search：このスロットに結合されたButtonPadがクリックされると、#dicNoに指定された辞書に対して#codeで指定された値を持つエントリーの検索を実行する。

#eMessage：このパッドの処理が出すメッセージを保持する。

#uDate：検索対象の辞書の更新日付を保持する。

#dicName：検索対象の辞書名を保持する。

#explan0：検索対象の辞書の説明テキストを保持する。このスロットにもパッド（StringHolderPV/PadModel）を接続する。

#codeList：検索対象の辞書のコードリストを保持する。

#dicList：EXFOR辞書に含まれるサブ辞書名のリストを保持している。

#dicNoList：EXFOR辞書に含まれるサブ辞書の辞書番号を保持している。

ビュー部ExforDictSearchPadVは、sendEvent:from:with:メッセージに反応し、次のように選択メニュを表示する。ニックネームが‘code’であるパッド（PromptPadWithSendView/PadModel）からのセレクトボタンが押されたイベントが送られた場合は、#codeListが保持するリストをメニュ表示し、選択された項目を#codeスロットに設定する。ニックネームが‘dicNo’であるパッド（NumberPadWithSendView/NumberPadModel）からのセレクトボタンが押されたイベントが送られた場合は、辞書名のリストをメニュ表示し、選択された辞書番号を#dicNoスロットに設定する。

8．PromptPadとNumberPadの機能拡張

NrdfDictToolPadとExforfDictSearchPadでは、辞書のタイプや検索するエントリーの見出し語コードをパラメータとして入力して、検索処理を行う。パラメータの特定の値を直接に入力するのであれば、PromptPadまたはNumberPadを該当スロットに接続して行えばよい。これらのパラメータは、パラメータの集合の中からメニューで選択して入力できることも必要である。このメニューの内容が他のパラメータの値によって決まるような場合は、どのパラメータの値を決めようとしているかが識別できなければならない。そこで、PromptPadとNumberPadのビュー部が<select>ボタンイベントを送れるように次

のように拡張している。

（1）PromptPadWithSend パッド

PromptPadView のサブクラスとして、PromptPadWithSendView を定義する。そして、pad-event action として redButtonDownAction メソッドを定義し‘sClicked’というイベントタイプを sendEvent:from:with:メッセージでおくる。PromptPadWithSend パッドとは、このように定義されたビュー部を持つ PromptPad の拡張である。

（2）NumberPadWithSend パッド

NumberPadView のサブクラスとして、NumberPadWithSendView を定義する。そして、pad-event action として redButtonDownAction メソッドを定義し‘sClicked’というイベントタイプを sendEvent:from:with:メッセージでおくる。NumberPadWithSend パッドとは、このように定義されたビュー部を持つ NumberPad の拡張である。

これにより、PromptPadWithSend パッド（PromptPadWithSendView/PadModel ）と NumberPadWithSend パッド（NumberPadWithSendView/NuberPadModelproperties）の#nickname スロットに適当な名前(接続されている親パッドのスロット名)を設定して子パッドとしてスロットに結合すれば、親パッドの View 部で、sendEvent:from:with:メソッドが呼び出されて‘sClicked’のイベントタイプとこのイベントを送ったパッドを識別できるようになる。このように、メニューからもスロットの値を決めることができる。

9．まとめ

CONTIP で利用しているデータベースサーバーは、CGI によりインターネットを介してアクセスできる。ここで記述した IntelligentPad のツールは、VisualWorks プラットフォームの Smalltalk 版で開発されている。したがって、インターネットにつながっている任意のクライアントマシンにこれらのパッドツールの配布が可能で、これらのパッドツールを用いればインターネット上のどのパソコンからでもこれらのデータベースをアクセスできるようにすることができる。

検索や保守のツールパッドは、それぞれ特定の用途に応じてカスタマイズまたは機能追加などの修正が可能である。

NRDF 辞書は、そのエントリーがこのデータベース上で更新されるであろう。このため、この辞書を入力ファイルと同一形式のテキストにバックアップするツールを開発する必要がある。

CONTIP のデータベース NRDBs には NRDF データに加え、NRDF と EXFOR の辞書データが加わったが、近い将来 EXFOR データが加わることになろう。そうなるとこれらのデータを統合しシームレスな利用環境を実現できるので、NRDF データと EXFOR データ間の相互変換、データベース全体を適当な切り口で可視化するツールの研究開発が可能となるであろう。

参考文献

[1] 長崎祥、田中譲「シンセティック・メディア・システム：IntelligentPad」コンピュータソフトウェア、Vol.11、No.1(1994)

[2] 田中譲「ミームメディア・アーキテクチャ IntelligentPad とその応用、情報処理、38巻3号(1997)

[3] 日本荷電粒子核反応データグループ、研究代表者：田中 一「荷電粒子核反応データファイル作成報告書」文部省科学研究費補助金特定研究、昭和56年3月

[4] 加藤幾芳「荷電粒子核反応データベース NRDF の現状とその利用」原子核研究、Vol.39、No.5(1995)

[5] V. McLane(ed.)「EXFOR MANUAL Center-to-Center Exchange Format」(1989)

付録１．　EXFOR 辞書に対するクラス定義

```
/* create EXFOR Dictionary Classes  */

CREATE CLASS ExforDict
(
 dicNo SMALLINT NOT NULL,
 dicName VARCHAR(22),
 uDate   DATE,
 explanation SEQUENCE VARCHAR(55),
 flag  CHAR(1),
 enpansion VARCHAR(55)
 );
CREATE CLASS Related AS SUBCLASS OF ExforDict
(

relInfo CHAR(1)
);

CREATE CLASS ExforDict1 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 1,
 code  VARCHAR(10)  NOT NULL,
 entrySeq INTEGER NOT NULL
 );

CREATE CLASS ExforDict2 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 2,
 code VARCHAR(10)  NOT NULL,
 expansion VARCHAR(25),
 keyReq    VARCHAR(4),
 codeReq   VARCHAR(6),
 relDic    VARCHAR(11),
 entrySeq  INTEGER NOT NULL
);

CREATE CLASS ExforDict3 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 3,
 code CHAR(7)  NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict4 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 4,
 code CHAR(1)  NOT NULL,
 expansion VARCHAR(4),
 relDic   SMALLINT,
 entrySeq INTEGER NOT NULL
 );

/* coutry -> area + country */
CREATE CLASS ExforDict5 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 5,
 code VARCHAR(6)  NOT NULL,
 country     CHAR(4),
 entrySeq INTEGER NOT NULL
);

/* institute -> area + country + institute */
CREATE CLASS ExforDict6 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 6,
 code VARCHAR(11)  NOT NULL,
 expansion VARCHAR(48),
 institute  CHAR(7),
 entrySeq INTEGER NOT NULL
 );

CREATE CLASS ExforDict7 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 7,
 code VARCHAR(10)  NOT NULL,
 entrySeq INTEGER NOT NULL
 );

CREATE CLASS ExforDict8 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 8,
 code VARCHAR(6)  NOT NULL,
 zNo  SMALLINT,
 elmSymbol  CHAR(2),
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict9 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 9,
 code VARCHAR(10)  NOT NULL,
 zNo   SMALLINT,
 elmSymbol CHAR(2),
 compound  CHAR(3),
 entrySeq INTEGER NOT NULL
);

/* Particles   */
CREATE CLASS ExforDict13 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 13,
 code VARCHAR(3)  NOT NULL,
 entrySeq INTEGER NOT NULL
);
```

```
CREATE CLASS ExforDict15 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 15,
 code CHAR(1) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict16 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 16,
 code VARCHAR(5) NOT NULL,
 relInfo   CHAR(1),
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict17 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 17,
 code CHAR(1) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict18 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 18,
 code VARCHAR(5) NOT NULL,
 category VARCHAR(55),
 entrySeq INTEGER NOT NULL
 );


CREATE CLASS ExforDict19 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 19,
 code VARCHAR(5) NOT NULL,
 category VARCHAR(55),
 entrySeq  INTEGER NOT NULL
);

CREATE CLASS ExforDict20 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 20,
 code VARCHAR(5) NOT NULL,
 entrySeq  INTEGER NOT NULL
);

CREATE CLASS ExforDict21 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 21,
 code VARCHAR(5) NOT NULL,
 category VARCHAR(55),
 entrySeq  INTEGER NOT NULL
);

CREATE CLASS ExforDict22 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 22,
 code VARCHAR(5) NOT NULL,
 category VARCHAR(55),
 entrySeq  INTEGER NOT NULL
);


CREATE CLASS ExforDict23 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 23,
 code VARCHAR(5) NOT NULL,
 category VARCHAR(55),
 entrySeq  INTEGER NOT NULL
);

CREATE CLASS ExforDict24 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 24,
 code VARCHAR(10) NOT NULL,
 category VARCHAR(55),
 checkFlag VARCHAR(1),
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict25 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 25,
 code VARCHAR(10) NOT NULL,
 category VARCHAR(55),
 dimension  VARCHAR(4),
 convFact    VARCHAR(11),
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict27 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 27,
 code VARCHAR(10) NOT NULL,
 zNo  SMALLINT,
 elmSymbol CHAR(2),
 aNo  SMALLINT,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict28 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 28,
 code VARCHAR(3) NOT NULL,
 entrySeq INTEGER NOT NULL
 );

CREATE CLASS ExforDict29 AS SUBCLASS OF ExforDict
```

```
(
 dicNo SMALLINT SHARED 29,
 code VARCHAR(3) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict30 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 30,
 code VARCHAR(3) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict31 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 31,
 code VARCHAR(5) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict32 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 32,
 code VARCHAR(3) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict33 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 33,
 code VARCHAR(3) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict34 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 34,
 code VARCHAR(3) NOT NULL,
 relInfo CHAR(1),
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict35 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 35,
 code VARCHAR(5) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict36 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 36,
 code VARCHAR(44) NOT NULL,
 dimension VARCHAR(4),
 resonance VARCHAR(1),
 category VARCHAR(55),
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict37 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 37,
 code VARCHAR(5) NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict42 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 42,
 code VARCHAR(3) NOT NULL,
 numCode SMALLINT NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict43 AS SUBCLASS OF ExforDict
(
 dicNo SMALLINT SHARED 43,
 code SMALLINT NOT NULL,
 entrySeq INTEGER NOT NULL
);

CREATE CLASS ExforDict50 AS SUBCLASS OF ExforDict(
 dicNo SMALLINT SHARED 50,
 code SMALLINT NOT NULL,
 entrySeq INTEGER NOT NULL
);
```