XENO: Computer-Assisted Compilation of Crossword Puzzles

P. D. Smith

Department of Computer Science, California State University, Northridge, 18111 Nordhoff Street, Northridge, California 91330, USA

XENO is a suite of programs which assist in the creation of crossword puzzles. It is an extension of a diagram filling program presented earlier.[1] Complete puzzles can be produced under the control of a small number of parameters, clues being selected from an indexed file. It is possible to generate thematic puzzles by including keywords in the input parameters. Use of a tree searching method results in a puzzle being generated efficiently.

1. INTRODUCTION

1.1 Background

An earlier program[1] produced filled crossword diagrams given a skeleton diagram and a dictionary of words. Diagrams processed were those characteristic of British rather than American puzzles. This program automated stage (ii) in the compilation of a crossword puzzle, the three stages being:

(i)   creation of a diagram with identified word slots
(ii) entry of letters into each unblocked square so that each word slot contains a single word or multi-word phrase
(iii) production of a clue for each word slot.

The XENO package contains programs which accomplish the second and third stages of puzzle production. In that the clues used are not generated by the program but rather selected from an existing file, XENO is best described as an aid to puzzle compilation rather than a puzzle compiler. Typical input to XENO is shown in Fig. 1 and corresponding output in Fig. 2. This paper describes the program which performs stage (iii) and notes enhancements made to the earlier stage (ii) program. Example XENO produced puzzles are included.

```
255
150
350
1
20
12
                + + +/
        + + +      /
            + + + +/
        + + +     +/
            +  ++ /
        + +  + +      /
              ++   + + /
        + +  +          /
      +          + + +  /
      + + + +         /
            + + +  /
      + + +        /
```

Figure 1. Typical input.

1.2 Aesthetics

Apart from technical problems in producing a puzzle there are aesthetic questions to be addressed. For example, what makes one selection of words and clues better than another? What are the characteristics of a good puzzle? The following factors will be considered in turn:

(a) selection of words and phrases
(b) interlocking of words
(c) quality of individual clues
(d) selection of clues
(e) cohesiveness of the puzzle.

(a) Kurzban and Rosen[2] present a list of types of words which should not occupy more than 10% of any puzzle. Their list includes dialect words, foreign words, and abbreviations. Restrictions on vocabulary are more important in American puzzles where, because of greater interlocking, a larger vocabulary is normally required. Clearly the same word should not appear more than once and, unless they are part of a puzzle's theme, closely related words should not appear. The vocabulary from which puzzle words are drawn is determined by the compiler's mental picture of a typical solver.

(b) Letters in intersecting squares should be of some help to the solver. Kurzban and Rosen[2] advise that they should be 'fairly common consonants or vowels, in positions they occupy less commonly'. For example B---R is recommended over S---E. Further, it is undesirable that a dictionary word could be produced by changing an unchecked letter in a solution.

(c) Macnutt[3] proposes that clues should be devised so that a solver should be in no doubt that a correct answer is indeed correct. It is unlikely that a program could generate challenging cryptic clues satisfying this requirement. The amount of real-world knowledge required would be enormous. An interesting possibility, however, is the use of the grammar devised by Williams and Woodhead[4] to generate a skeletal clue form. This would guide a human compiler in producing clues of forms they identify as being under-used.

(d) Kurzban and Rosen suggest that half the selected clues be of forms reasonably easy to solve (e.g. multiple definitions and simple charades involving anagrams) and that there be clues of this type in all parts of the puzzle. Subject to this requirement it seems desirable to have clues of a uniform difficulty.

```
!1 !   !2 !   !3 !   !4 !+++!5 !+++!6 !+++!
! s ! p ! l ! u ! r ! g ! e !+++! s !+++! h !+++!
!   !   !   !   !   !   !   !+++!   !+++!   !+++!
!---!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!   !+++!   !+++!7 !   !   !   !   !   !
! y !+++! i !+++! o !+++! d ! a ! h ! l ·! i ! a !
!   !+++!   !+++!   !+++!   !   !   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!8 !   !   !   !   !+++!   !+++!   !+++!   !+++!
! n ! y ! l ! o ! n !+++! g !+++! o !+++! l !+++!
!   !   !   !   !   !+++!   !+++!   !+++!   !+++!
!---!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!   !+++!   !+++!9 !   !   !   !   !+++!
! d !+++! a !+++! d !+++! i ! d ! e ! a ! l !+++!
!   !+++!   !+++!   !+++!   !   !   !   !   !+++!
!---!---!---!---!---!---!---!---!---!---!---!---!
!10 !   !   !11 !   !   !   !+++!   !+++!+++!12 !
! r ! a ! c ! c ! o ! o ! n !+++! h !+++!+++! c !
!   !   !   !   !   !   !   !+++!   !+++!+++!   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!+++!   !+++!+++!13 !   !   !   !   !   !
! o !+++!+++! l !+++!+++! g ! r ! o ! c ! e ! r !
!   !+++!+++!   !+++!+++!   !   !   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!14 !   !   !   !15 !+++!+++!   !+++!+++!   !   !
! m ! o ! s ! a ! i ! c !+++!+++! r !+++!+++! o !
!   !   !   !   !   !+++!+++!   !+++!+++!   !   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!+++!   !+++!16 !   !17 !   !18 !   !   !
! e !+++!+++! r !+++! r ! a ! i ! n ! b ! o ! w !
!   !+++!+++!   !+++!   !   !   !   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!+++!19 !   !   !   !   !+++!   !+++!   !+++!   !
!+++! a ! r ! i ! s ! e !+++! d !+++! a !+++! f !
!+++!   !   !   !   !   !+++!   !+++!   !+++!   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!+++!   !+++!   !+++!   !+++!20 !   !   !   !   !
!+++! j !+++! n !+++! a !+++! i ! g ! l ! o ! o !
!+++!   !+++!   !+++!   !+++!   !   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!21 !   !   !   !   !   !+++!   !+++!   !+++!   !
! p ! a ! r ! e ! n ! t !+++! o !+++! s !+++! o !
!   !   !   !   !   !   !+++!   !+++!   !+++!   !
!---!---!---!---!---!---!---!---!---!---!---!---!
!+++!   !+++!   !+++!22 !   !   !   !   !   !   !
!+++! x !+++! t !+++! e ! n ! t ! r ! a ! n ! t !
!+++!   !+++!   !+++!   !   !   !   !   !   !   !
```

Across

1 Swell outside—endless play makes extravagant show (7)
7 Roald, back in top form, blooms colourfully (6)
8 New York and half London makes synthetic fibre (5)
9 Youth that is, destroyed model (5)
10 Source of Davy Crockett's cap (7)
13 Negro certainly conceals shopkeeper (6)
14 Surface decorated with small coloured pieces of material (6)
16 'Somewhere over the —, way up high' (E. Y. Harburg) (7)
19 Czar is easily containing move upward (5)
20 Big loon hides snow dome (5)
21 'Wealth is the,— of luxury and indolence' (Plato) (6)
22 One who entered, ran inside damaged tent (7)

Down

1 China —: 1979 movie (8)
2 — Garden: 1936 ballet by Antony Taylor (5)
3 Pole in door upset musical form (5)
4 Trimming is seen in fixed gin game (6)
5 Ho Ron! (He's crazy but useful for putting on shoes) (8)
6 Rounded elevation in the landscape (4)
11 Single reed woodwind instrument (8)
12 Plant also known as buttercup (8)
15 Make rodent hold string in church (6)
17 Simpleton (5)
18 Very light wood (5)
19 Greek hero of the Iliad (4)

Figure 2. Example output from input 1.

(e) A measure of puzzle cohesiveness can be achieved by having a common theme in either the diagram words or the clues. Cross-references from one clue to another also help produce an integrated puzzle. An example due to 'Araucaria' is

It's a 21 24 giving a smooth finish (5, 5)

The solutions to clues 21 and 24 were NEAR and THING yielding CLOSE SHAVE as the answer.

1.3 Literature survey

There are few published programs in this field. Those surveyed in Ref. 1 do not go beyond the diagram filling stage.

'Crossword Magic'[5] aids a user in preparing a puzzle; it handles the problem of interlocking and allows a user to move words round the diagram. The diagram, initially completely blocked, evolves as words are entered. This program is intended for interactive generation and solving on microcomputers with a single disc drive. It has no dictionary, so words and clues are user supplied as required.

A commercial puzzle-producing company (Wellingtons) decided not to go ahead with automated puzzle production after a pilot program. They found that diagram-filling times were not competitive with those of professional compilers.[6]

2. OVERVIEW

2.1 Clues and parameters

Each entry for the dictionary used in diagram filling points to a list of clues held in a file. Each clue has associated with it:

(i) a difficulty rating—an integer from 1 to 100 which represents a subjective estimate of the difficulty of the clue when taken in isolation from the rest of the puzzle
(ii) a vector representing a rudimentary classification for the clue (see Section 4).
(iii) the text of the clue which includes the implied division of the hidden or target string. For example both the following

The villain of the Rape of Lucrece (3, 6)
One who practices therapy (9)

might be clues for THERAPIST.

In order to produce a filled diagram together with a set of clues, XENO requires the skeleton diagram and five parameters:

(1) clue-type vector   (CLUVEC)
(2) minimum total puzzle difficulty   (MINTOT)
(3) maximum total puzzle difficulty   (MAXTOT)
(4) minimum individual clue difficulty   (MININD)
(5) maximum individual clue difficulty   (MAXIND)

When a filled diagram is produced, a table (SLOTTABLE) is also generated which contains information about each word slot. It is a simple matter to present selected clues in the conventional form shown in Fig. 2.

2.2 Aesthetic considerations

Several measures designed to produce pleasing puzzles were implemented; the five factors in Section 1.2 have been considered.

(a) Dictionary words are tagged if they fall into one of the marginally acceptable categories. Of the words in the current dictionary between 2 and 3% are so marked. In the diagram filling stage, a tagged word is rejected if its inclusion would cause the total of such words to exceed 10% of the word slots. Because of the scattered manner in which the diagram is filled, marginally acceptable words should be well spread. Although no word can appear more than once, there is currently no check on the use of closely related words, such as those with the same root form.

(b) There are no controls at present on letters which occupy interlocking squares. An analysis of the dictionary will be required to determine which positions particular letters occupy commonly and which less commonly.

(c) Clues currently in use have been created by the author purely for testing purposes as a more interesting alternative to dummy strings. No claims of elegance or fairness are made for those clues appearing in the examples.

(d) The CLUVEC parameter enables a user to specify the types of clue to be used. There is as yet no means of specifying the required percentage for each type or of ensuring a spread for a particular type.

(e) Theme puzzles can be generated. Figure 3 shows how a theme is specified and Figure 4 is a resulting puzzle. XENO can accept partially filled diagrams as input thus allowing a user to fill in key theme words.

After making the clue selection the texts are edited and each occurrence of a diagram word is replaced by an appropriate reference. Three examples of this replacement can be seen in Fig. 4. In order to increase the frequency with which replacement is performed it would be necessary to direct clue selection to favour clues containing diagram words. It is unlikely that the extremes of Ref. 7 would be achieved where sixteen clues refer to 23 down.

The effects of the user-supplied parameters on the diagram filling stage are discussed in Section 3. When a filled diagram is produced, a process of list formation

```
255
80
450
1
50
11
+ +            +       /
  +   +   +   +   +   /
      +               /
  +   +   +   +   +   /
          +           /
+ +   +   +   +   + +/
      +               /
  +   +   +   +   +   /
          +           /
  +   +   +   +   +   /
      +           + +/
      soccer, baseball
```

Figure 3. Typical input (with keywords).

```
!+++!+++!!1  !   !2  !   !3  !+++!4  !   !5  !
!+++!+++!  r ! a ! c ! e ! s !+++! m ! a ! l !
!+++!+++!   !   !   !   !   !+++!   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!
!6  !+++!   !+++!   !+++!   !+++!   !+++!   !
! d !+++! o !+++! u !+++! a !+++! e !+++! e !
!   !+++!   !+++!   !+++!   !+++!   !+++!   !
!---!---!---!---!---!---!---!---!---!---!---!
!7  !   !   !+++!8  !   !   !   !   !   !   !
! a ! r ! t !+++! r ! e ! f ! e ! r ! e ! e !
!   !   !   !+++!   !   !   !   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!   !+++!   !+++!   !+++!   !+++!   !
! s !+++! a !+++! v !+++! e !+++! s !+++! d !
!   !+++!   !+++!   !+++!   !+++!   !+++!   !
!---!---!---!---!---!---!---!---!---!---!---!
!9  !   !   !   !   !   !+++!10 !   !   !   !
! h ! i ! t ! m ! e ! n !+++! r ! e ! d ! s !
!   !   !   !   !   !   !+++!   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!
!+++!+++!   !+++!   !   !+++!11 !+++!   !+++!+++!
!+++!+++! o !+++! s !+++! c !+++! y !+++!+++!
!+++!+++!   !+++!   !+++!   !+++!   !+++!+++!
!---!---!---!---!---!---!---!---!---!---!---!
!12 !   !   !   !+++!13 !   !   !   !   !14 !
! f ! o ! r ! d !+++! r ! o ! b ! s ! o ! n !
!   !   !   !   !+++!   !   !   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!   !+++!15 !+++!   !+++!   !+++!   !
! i !+++! c !+++! s !+++! o !+++! i !+++! i !
!   !+++!   !+++!   !+++!   !+++!   !+++!   !
!---!---!---!---!---!---!---!---!---!---!---!
!16 !   !   !   !   !   !+++!17 !   !   !   !
! f ! o ! u ! l ! t ! i ! p !+++! d ! e ! n !
!   !   !   !   !   !   !+++!   !   !   !   !
!---!---!---!---!---!---!---!---!---!---!---!
!   !+++!   !+++!   !+++!   !+++!   !+++!   !
! e !+++! f !+++! e !+++! e !+++! e !+++! e !
!   !+++!   !+++!   !+++!   !+++!   !+++!   !
!---!---!---!---!---!---!---!---!---!---!---!
!18 !   !   !+++!19 !   !   !   !   !+++!+++!
! r ! e ! f !+++! p ! a ! r ! i ! s !+++!+++!
!   !   !   !+++!   !   !   !   !   !+++!+++!
```

Across

1 Troubled fright runs for pennants (5)
4 Big —, roving soccer manager (3)
7 She leaves Edinburgh club painting for example (3)
8 Jack Taylor's role in 1974 World Cup Final (7)
9 Every baseball club needs these assassins (3, 3)
10 Team found in Manchester and Cincinatti (4)
12 Whitey —, Yankee pitcher in the 1960s (4)
13 Deprive male offspring, make England manager (6)
16 Snick at the plate—bad advice (4, 3)
17 Home ground of Millwall F.C. (3)
18 Short official in centre field (3)
19 City where 5 played in European Cup Final (5)

Down

1 Vulnerable part of pitcher's shoulder (7, 4)
2 Assets for pitcher and model (6)
3 Secure, welcome call for base runner (4)
4 Liverpool, Everton etc. keep river in bounds (6, 5)
5 Sounds as though this club is in front (5)
6 Lasorda show contains sudden rush (4)
11 Defender for England and 5 in the 1970s (6)
12 If 18 confused turns to Perthshireman? (5)
14 Number of innings in a regular baseball game (4)
15 Goalkeeper has to watch his, limit of four (4)

Figure 4. Example output from input 2.

followed by a tree search is used to establish a difficulty rating for each word slot. Individual clues are then selected; Section 5 describes how this is done. Section 6 details the current implementation, Section 7 describes the results of some XENO runs.

3. DIAGRAM FILLING

We wish to exclude from a filled diagram words for which no suitable clue exists. Their inclusion would result in failure to compile a puzzle. At the same time any checking process should be relatively fast, because several hundred words may have to be screened during diagram filling.

A compromise solution is implemented. Summary information about the clues held for a word is stored in its dictionary entry. When a word is being considered for inclusion in the diagram the largest and smallest difficulty values are compared with MININD and MAXIND. The clue type summary field is also compared with CLUVEC.

If there are keywords in the input parameters, a marking operation takes place before diagram filling. Dictionary words having at least one clue tagged with any of the given keywords are marked in a simple way. The clues themselves are similarly marked. Only marked words are considered when filling the diagram.

There are ways in which a word can slip through this screening process because clue type and difficulty are independent. XENO would then fail during clue selection. This problem, which has not been a serious one, will probably become rarer as the clue collection grows.

4. CLUE CLASSIFICATION

Arnot,[8] in a discussion of the development of cryptic puzzles, identifies a number of clue categories. These were used as the basis for the classification scheme in XENO. Table 1 contains examples of the eight categories used. The general knowledge clues (first two categories) have only one possible answer. Cryptic clues (the remaining six categories) have potentially several answers. The cryptic examples are from puzzles by 'Araucaria' published in recent editions of the *Manchester Guardian Weekly.*

Some clues are suitable for words running in a particular direction. For example consider the following clues for ENID.

> This girl seems to be eating backwards
> Eat up, girl!

Table 1. Clue categories

| Category	Example	Answer
Quotation	'To be or—to be (3)' Hamlet	NOT
Other general knowledge	Largest state in the U.S.A. (6)	ALASKA
Hidden word	Some of America's glory (4)	FAME
Anagram	Tried by sad eyes (7)	ESSAYED
Second definition	Two pieces of meat quick (4, 4)	CHOPCHOP
Charade	Girl sees America from pole to pole (5)	SUSAN
Homophone	Tree that sows, say (5)	CEDAR
Other cryptic	Did Mao think that April Fool's Day would never come? (4, 5)	LONGMARCH
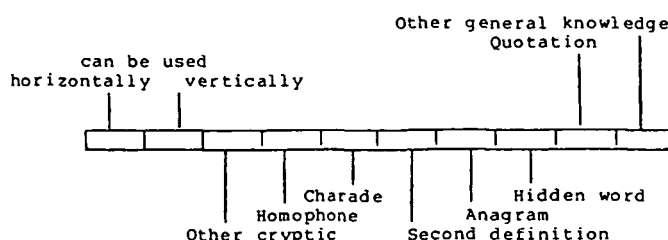
Figure 5. Coding scheme for clues.

The first (from *The Times,* 1 February 1930) clued ENID in a horizontal slot whereas the second (from the *Manchester Guardian Weekly,* 28 January 1982) was used when the word ran vertically.

Clues can be in more than one category; for example, 'Sort of green (5)' [GENRE] is both an anagram and a second definition. Because of this, and the need during diagram filling to be able to compare CLUVEC with a representation of the types of all clues present in a set, the context free coding scheme shown in Fig. 5 is used to produce a bit vector for a clue. Thus 'Sort of green' would have associated with it 1100011000 (792 decimal). The bit vector stored with a word is the union of the individual vectors for its clues.

Bits in the CLUVEC parameter are set using the same scheme. CLUVEC indicates acceptable clue types; a decimal value of 252 (11111100) would result in a puzzle containing only cryptic clues.

An extension would be the recording of author identifiers. Different clue composers have different styles and conventions (see Refs 3 and 8). Mixing clues from different authors in the same puzzle is unlikely to make it easier to solve.

5. CLUE SELECTION

Clue selection is divided into six phases:

(1) screening and forming lists
(2) ordering of table entries
(3) computing cumulative totals
(4) random list reordering
(5) tree searching
(6) selecting randomly from clue lists.

Phases (1) to (5) attempt to establish a difficulty rating for each word in the diagram such that (a) there is, for that word, at least one clue with that rating which satisfies input constraints, and (b) the total rating for the puzzle (sum of the individual ratings) is within the MINTOT and MAXTOT bounds.

Phase 6 selects a clue for each word from among the eligible clues with the rating determined in the first 5 phases. Details of the six phases follow.

(1) Dictionary entries, apart from a pointer to a list of clues, hold information about those clues: range of difficulty ratings and clue types. For each SLOTTABLE entry a list structure is formed of pointers to suitable clues.

(2) A clue rating for each word position is determined using a depth first tree search. Before the search SLOTTABLE is ordered by number of different ratings. In this way entries with fewest degrees of freedom come first and backtracking is minimized.

Table 2. SLOTTABLE entries during clue selection

mosaic	1	465	287
lilac	1	455	277
raccoon	2	435	257
splurge	2	415	242
dahlia	2	395	226
balsa	2	375	211
rondo	3	355	196
parent	3	335	186
ideal	3	320	178
create	3	300	166
shoehorn	3	280	158
rainbow	3	260	146
entrant	3	240	134
ajax	3	220	124
igloo	3	200	114
crowfoot	3	180	104
syndrome	4	160	89
idiot	4	140	74
edging	4	120	62
hill	4	100	52
nylon	4	80	42
grocer	4	60	32
arise	5	40	22
clarinet	5	20	10

(3) In order that the tree search can back up as soon as a partial solution can no longer lead to a complete one, cumulative totals are computed for the maximum and minimum entries. Table 2 shows the table for the examples of Figs 1 and 2 after phases (2) and (3) were performed.

(4) Each list structure is randomly reordered. This randomization ensures that different puzzles can be produced from the same filled diagram and input parameters.

(5) The tree search tries to establish a rating for each word slot so that the total is within MAXTOT and MINTOT bounds. If no solution can be found, diagnostics are printed.

(6) A random number generator is used to select elements from the appropriate sublists of pointers.

6. CURRENT IMPLEMENTATION

XENO is implemented in OMSI Pascal-1[9] running on a PDP-11/70 under RSTS/E. Pascal-1 extends the Pascal language[10] to permit random access to files. A file can be treated as an array of records.

Apart from the puzzle generating programs, XENO has a number of utilities which operate on the file structure to dump, edit and add words and clues.

The current dictionary contains 14 000 words, 6000 more than that used by the earlier program.

7. RESULTS

Puzzles of varying difficulty have been generated. Times required to compile a puzzle varied significantly. XENO was run 8 times with the diagram of Fig. 1 using different

seeds for the random number generator. In 4 cases a puzzle was compiled in about 90 s, in 2 cases it took approximately 10 min, in another case about 25 min, and the final run was stopped after 2 h (most of which was spent in trying to find a combination of words to fit the final 3 slots). It is desirable to impose a relatively short time limit on XENO runs and, if it fails to complete a puzzle in that time, have it output a partial one.

Largely because there are as yet few clues held for any particular word, the process of finding a clue set, given a filled diagram, is fast even in cases where the input parameters are such that there are few sets possible.

Across

1 Ostentatious demonstration or effort (7)
7 A plant bearing large brilliantly coloured flowers (6)
8 Hard-wearing synthetic material developed in the 1930s (5)
9 Mental image of perfection (5)
10 American ringtailed animal, often builds den in tree (7)
13 A dealer in food supplies (6)
14 Surface decorated with small coloured pieces of material (6)
16 'Somewhere over the —, way up high' (E. Y. Harburg) (7)
19 'I will — and go now, and go to Innisfree' (Yeats) (5)
20 Eskimo snow house (5)
21 Father or mother (6)
22 One who enters, into a competition for example (7)

Down

1 Group of symptoms characterizing some condition (8)
2 — Garden: 1936 ballet by Antony Taylor (5)
3 Musical form in which digression and return alternate (5)
4 Border, of plants or lace for example (6)
5 Device aiding putting on of shoes (8)
6 'Home is the sailor, home from the sea, and the hunter home from the —' (Stevenson) (4)
11 Single reed woodwind instrument (8)
12 Herbaceous plant of genus Ranunculus (8)
15 'There will be a time to murder and —' (Eliot) (6)
17 The —: Dostoyevsky novel (5)
18 Very light wood (5)
19 Son of Telamon in Greek legend (4)

Figure 6. Partial output from input 1.

Across

1 Wild gulpers make ostentatious demonstration (7)
7 In modern era, confused salute garden bloom (6)
8 Pole only damaged synthetic fibre (5)
9 Youth that is, destroyed model (5)
10 Upset drink in navy make American animal (7)
13 Shopkeeper ordered to expand we hear (6)
14 Coma is disorder of slaves' leader (6)
16 Multicoloured deity at front of ship (7)
19 First word to knight? (5)
20 XENO sizes ice house? Not what I hear (5)
21 Standard damaged snare gets progenitor (6)
22 One who entered, ran inside damaged tent (7)

Down

1 Symptom set sounds like vice centre (8)
2 One bred out of the dead land in cruel April (5)
3 Alternatively, put on new musical form (5)
4 Poor bell sound in, for example, border (6)
5 Footwear aid shaved round broken garden tool (8)
6 One of seven under Rome? (4)
11 Woodwind confused horn in charge (8)
12 Rot of cow damaged buttercup (8)
15 Make rodent hold string in church (6)
17 Simple minded five hundred, and one in endless trifle (5)
18 Upending a slab of modelling wood (5)
19 Greek hero of German assent in American chopper (4)

Figure 7. Partial output from input 1.

Table 3. Summary of three examples

		Fig. 2	Fig. 6	Fig. 7
CLUVEC		255	3	252
MINTOT		150	0	375
MAXTOT		350	500	600
MININD		1	1	20
MAXIND		20	40	60
Across	1	16	20	20
	7	20	15	25
	8	10	15	20
	9	20	12	20
	10	20	15	25
	13	10	15	20
	14	10	10	25
	16	15	15	20
	19	12	15	20
	20	10	15	20
	21	12	8	25
	22	20	10	20
Down	1	15	17	20
	2	20	20	25
	3	15	10	20
	4	10	12	20
	5	15	12	20
	6	10	15	20
	11	10	10	30
	12	17	25	20
	15	20	18	20
	17	12	15	20
	18	15	15	20
	19	15	25	25
Total		349	359	520

Typically less than 30 s of processor time was required by this stage for each of the four examples given in this paper. Figures 6 and 7 show other puzzles generated from the diagram of Fig. 2. Table 3 gives, for the three examples using the diagram, the values of the other parameters, ratings for the individual clues in the generated puzzle and the total puzzle difficulty.

Figure 6 is intended to be a general knowledge puzzle with practically no restrictions on clue choice. Contrast this with the clues shown in Fig. 7, which represent a cryptic puzzle more difficult than that of Fig. 2. As can be observed from Table 3, the individual clues of Fig. 7 are required to be at least as difficult as those of Fig. 2 and the puzzle as a whole is required to be more difficult.

Figure 4 shows a puzzle generated when Fig. 3 was the input to XENO. Only clues which had been tagged with 'soccer' or 'baseball' (or both) were candidates in this case and only words with such clues could appear in the diagram. Some post-processing of the clues has taken place. References to 5 down have replaced 'Leeds' in the clues to 19 across and 11 down; 'ref' in the clue to 12 down has been replaced by '18'. It is unlikely that the latter substitution would be made by a human compiler because it obscures the anagram. The use of both 'ref'

and 'referee' in the same diagram is undesirable (see Section 2.2).

8. CONCLUSIONS

XENO is a feasible way of producing puzzles which appear to have been compiled by hand. The parameter mechanism together with the clue classification and keyword schemes have proved very flexible in generating a variety of puzzles.

XENO can produce a puzzle considerably faster than the author working with pencil and paper. It is probable that it can produce one significantly faster than a professional compiler.

9. FUTURE DEVELOPMENTS

Enhancements are required before the puzzles produced by XENO are of generally publishable quality. The deficiencies noted in Sections 2.2 and 4 need to be rectified.

Further degrees of automation are possible. XENO could be rounded out with the generation of skeleton diagrams. A user should be able to specify the size of the diagram and have one generated which satisfies conventions on number of slots, interlocking and symmetry. Human compilers may create the diagram and fill in the words in parallel (see for example Ref. 3, Chapter IX). Enabling XENO to modify the diagram dynamically (while still preserving symmetry) may result in shorter compilation times.

In producing a series of puzzles where it is undesirable to have the same word occurring more than once, XENO could be made to exclude from consideration words in a given list.

Experiments with automatic generation of quotation clues would be possible given a local implementation of the CLOC package.[11,12] Data structures used by CLOC are particularly useful; they enable quotations of arbitrary length to be generated including any occurrence of an arbitrary word in a text. As a further aid to clue construction a program could be devised which, given a word or phase, identifies anagrams and hidden words.

A long-term goal is the production of puzzles in the style of particular human compilers. This will require analyses of vocabulary and clue style in published puzzles before any synthesis is possible. Generated puzzles could range from those that Macnutt[3] characterizes as time fillers, typically requiring an average vocabulary and no reference books, to 'Torquemada' type puzzles which may take a considerable time to solve.

REFERENCES

1. P. D. Smith and S. Y. Steen, A Prototype Crossword Compiler. The Computer Journal 24, 107 (1981).

2. S. Kurzban and M. Rosen, The Compleat Cruciverbalist, Van Nostrand Reinhold, New York (1981).

3. D. S. Macnutt, *Ximenes on the Art of the Crossword*, Methuen, London (1966).
4. P. W. Williams and D. Woodhead, Computer assisted analysis of cryptic crosswords. *The Computer Journal* **22**, 67 (1979).
5. S. Sherman, *Crossword Magic*, available from L & S Computerware, 1589 Fraser Drive, Sunnyvale, California 94087.
6. K. J. Miller, private communication (October 1982).
7. 'Araucaria', Crossword. *Manchester Guardian Weekly* **126** (No. 11), (March 1982).
8. M. Arnot, *What's Gnu: a history of the Crossword Puzzle*, Vintage Books, New York (1981).
9. Oregon Software, *Pascal-1 Language Specification*, Oregon Software Inc., 2340 S. W. Canyon Road, Portland, Oregon 97201 (1980).
10. K. Jensen and N. Wirth, *Pascal: User Manual and Report*, Springer, Berlin (1974).
11. A. Reed, CLOC: A collocation package. *Association for Literary and Linguistic Computing Bulletin* **5** (No. 2), (1977).
12. A. Reed, CLOC—A general-purpose concordance and collocation generator, in *Advances in Computer-aided Literary and Linguistic Research*, edited by D. E. Ager, F. E. Knowles and J. Smith, University of Aston (1978).