LEARNING RACQUETBALL BY CONSTRAINED EXAMPLE GENERATION*

By Leonard P. Wesley

Department of Computer and Information Science
University of Massachusetts
Amherst, Mass. 01003

ABSTRACT

A model of learning in a highly active and competitive environment is presented here. This paper posits that learning, under these conditions, can be characterized as accommodating examples produced by the Constrained Example Generation (CEG) process. The role of the"CEG process in the LRCEG system is demonstrated by a typical learning scenario.

I INTRODUCTION

This paper Introduces an emerging system designed to learn racquetball by a process of constrained example generation (LRCEG). Having the capacity to generate a set of constrained examples has been shown to be essential in many domains [1]. [2]. Furthermore, learning as a form of accommodating and assimilating knowledge has been demonstrated to be useful to a number of systems [6]. .The implication here is that some forms of learning can be modeled, in part, as a process of retrieval, modification, or construction of domain specific knowledge, represented as examples C33.C53. Exploring a variety of learnirg issues within the RB domain is made possible by the generality and robustness of the CEG paradigm [4],

Racquetball (RB) as a learning domain has some advantages. The speed and competitive nature of the aport facilitates investigating learning under rapidly changing and strenous conditions. Many important occupations require learning the appropriate behavior for unpredictable, fast-paced, and demanding environments (e.g., medicine, athletics, aircraft piloting). The learning required to perform tasks in these domains can be very different (although not mutually exclusive) from a typical classroom education. The generality and robustness of the CEG paradigm allows exploring a variety of learning-RB issues [4].

II ACQUISITION AND REPRESENTATION OF DOMAIN SPECIFIC KNOWLEDGE

Some knowledge about RB must be obtained and represented before the LRCEG system can attempt learning. Knowledge for the initial set of six examples was extracted from interviews of a RB expert.

•This work supported, in part, by free computer time from the COINS department at the University of Massachusetts, Amherst, Mass. 01003.

The syntax of the expert's information was "IF A PLAYER ENCOUNTERS object-of-encounters THEN TRY AND MEET object-of-meet AND RESPOND object-of-respond.". The context within which learning occurs is the 'object-of-encounters', and the appropriate behavior for meeting and hitting the ball is 'object-of-meet' and'object-of-respond' respectively. An example of the expert's advice is "IF A PLAYER ENCOUNTERS a fast cross-court shot THEN TRY AND MEET the ball waist high AND RESPOND with a ceiling shot." Both the context and behavior knowledge is translated into velocity and RB court X,Y,2 coordinates (i.e., 'coordinate-velocity' information). This knowledge is represented in a frame-like data structure [Fig 1a,b3.

```
(CONTEXT (SEMANTICS (X dX) (Y dY) (Z dZ) (V dV))
  (OBJECTIVE objective-function-definition)
  (SITUATION situation-function-definition)
  (RESPONSE response-info-4-furction-definition))
                    (a.)

(long-term-success-rate X Y Z V)
(X$ Y$ Z$ V$)
(short-term-success-rate response-function-def)
                    (b,)
```

Fig 1

(a.) This figure represents the LISP data structure used to represent the examples. X,Y,Z, and V are the coordinates and velocity of the ball. dX,dY,dZ, and dV specify the allowable variance of X,Y,Z, and V respectively.

(b,) An expansion of a portion of Fig 1a. X,Y,Z and V specify the coordinates and velocity of where to return the ball. X$, Y$, Z$, and V$ specify the maximum amount the modification process may change X.Y.Z. and V.

CONTEXT, in Fig 1a, is the context or example name (e.g., cross-court). The values of the SEMANTICS, OBJECTIVE, and RESPONSE slots are the corresponding domain and 'coordinate-velocity' information.

III THE LRCEG SYSTEM

The LRCEG system is written in LISP and runs on a VAX 11/780. A flow diagram of the LRCEG system and how a user interacts with it appears in Fig 2. System operation can be described in thirteen steps (corresponding to box numbers in Fig 2.) as follows: (1)Initial ball X,Y,Z position and velocity V, number of iterations, and discrete time interval is specified by the user. (2)A snapshot (instantaneous ball position and velocity) information is gathered. (3)Retrieve example(s) (from box 14) with satisfiable semantic constraints(i.e. X,Y,Z,V from snapshot are within dX,dY,dZ,dV of X,Y,Z,V from semantics slot of the example. Retrieval output is a set of examples that are candidates for modification. (4)Modification involves calling the function-definition programs defined in the OBJECTIVE, SITUATION, and RESPONSE slots (see Fig 1a.). The objective program modifies the X,Y,Z coordinates of where a player meets the ball, The player computes the trajectory of the ball and modifies where, on the RB court, to be in order to satisfy the constraint of meeting the ball at the position specified by the "object-of-meet". The situation function quantifies the players "situation" which is inversely proportional to the distance between the player and the objective. The response function modifies the X.Y.Z coordinates of where to hit the ball and computes the short-term-success-rate which is a "goodness" measure of the example's response specification. Examples are modified if improvement can be made in the short-term-success-rate. Each example is trying to obtain the highest "goodness" (i.e., short-term-success-rate value) measure possible, in effect competing with other examples to be selected as the one having the best response. A better situation, higher long-term-success-rate, slower ball, and faster player will tend to increase the short-term-success-rate. All examples are sorted on their short-term-success-rate before step 5. This allows subsequent retrieval, modification, and execution of the currently best examples first. Being able to retrieve and manipulate the best examples first is important if time and quantity constraints are placed on the learning process. (5)Execute action specified by objective slot of currently best example (i.e., try to meet the ball by updating current X,Y,Z of player). (6)Iterations completed or player has hit the ball. (7)Update ball X,Y,Z and V for next iteration. (8)Execute the action specified in the response slot by returning the example with the currently highest short-term-response-rating. (9)An example has failed if executing its response results in a missed or skipped ball (hits floor before front wall). Otherwise, threshold the response rating of the returned example to determine the success or failure of its response. (10)Self explanatory (11)Change the initial context slightly (i.e., vary the initial ball position and/or velocity). (12)Compute the statistical success rate of the example(s) and use a sigmoidal function to update long-term-success-rate of the example from 9. (13) Replace old long-term-success-rate with updated one from 12 and add the modified example to 14. This step is the accommodation process as the new example is now available for competition and evaluation with old examples in subsequent environments.

The LRCEG system is learning successfully if it returns a example having the best response rating for a previously similar context (i.e., what might work best in the future is what previously worked best in similar contexts).

IV EXPERIMENTAL RESULTS

This section presents Figs, 3 through 7 as a typical learning scenario in which the system ('player-l') learns the best response for a cross-court shot.

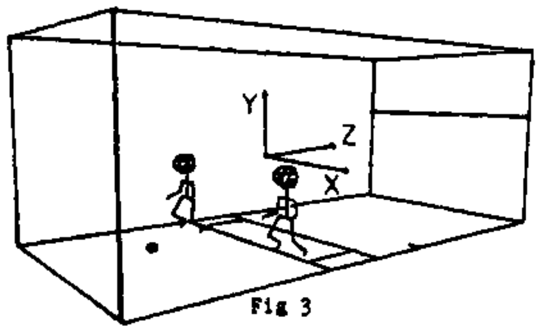

Fig 3

Illustrates a typical output of the display component of the LRCEG system. The figure also displays a typical snapshot along with the object axes.

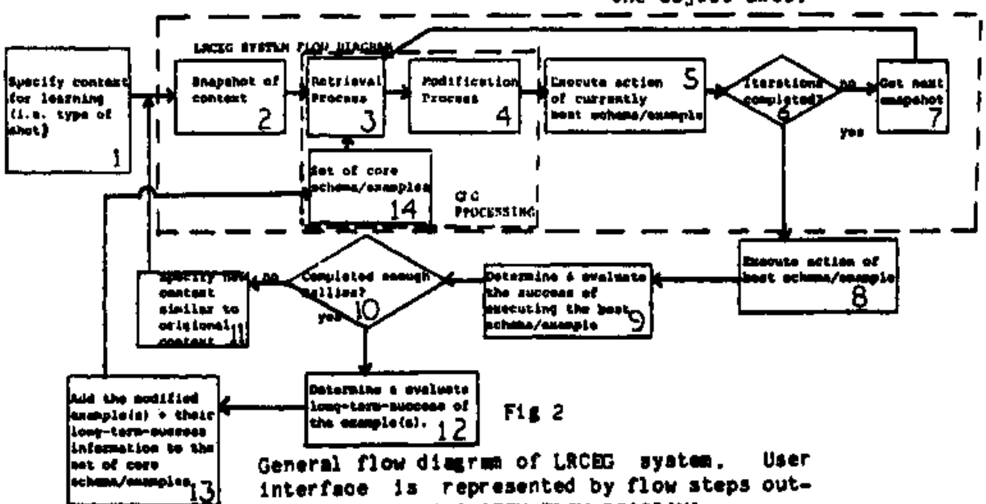

Fig 2

General flow diagram of LRCEG system. User interface is represented by flow steps outside the 'LRCEG SYSTEM FLOW DIAGRAM' area.

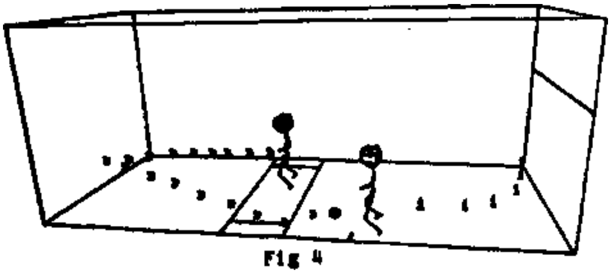

Fig 4

After rotating Fig 3 +45 degrees about the y axis. '1' traces the path of 'player-1'. *B' traces the trajectory of the ball start* ing at 'player-2' who does not move in this scenario. For display purposes the position of the ball and players are not indicated for every iteration.

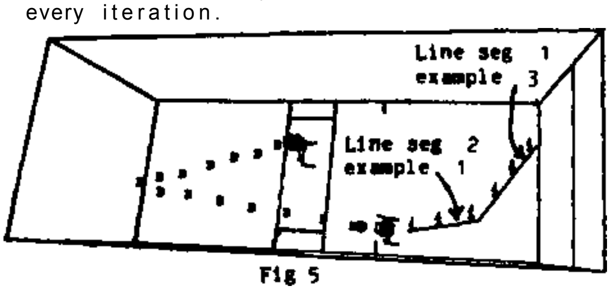

Fig 5

Sane as Fig 4 except rotated -90 degrees about the z axis. Line segment 1 illustrates 'player-r executing the action specified by example number 3. Line segment 2 intersects line segment 1 at the point where example 1's short-term-success-rate is higher than example 3's. At the intersection point *player-l' begins executing the action specified by example 1. When 'player-1* meets the racquetball example 1 specifies where to hit it.

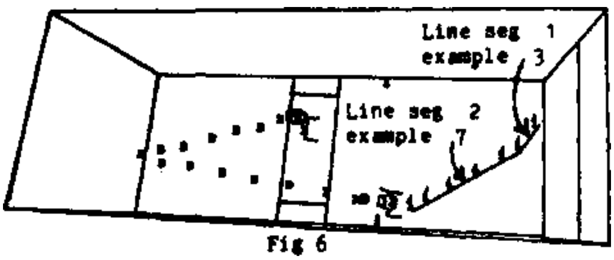

Fig 6

A new and seperate example 7 has been created and accommodated as a result of steps 12-14 in Fig 2. Except for the long-term-success-rate, information for example 7 is a copy of the latest modifications) to example 1. Example 7's long-term-success-rate was computed to be .75

The parent of example 7 (i.e., example 1) now has a long-term-success-rate of .55 Example 3's long-term-success-rate was reduced to .45 . (NOTE: initial long-term-success-rates for all examples was •5). This figure shows how example 7 begins influencing player-Ts behavior earlier in a similar rally. Examples 2 and 4-6 did not influence player-Ts behavior because their semantic constraints were not satisfied and/or their short-term-success-rate was not high enough.

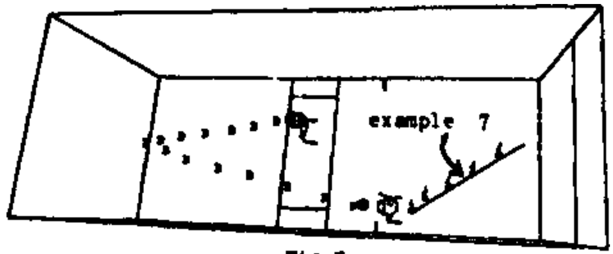

Fig 7

Finally the long-term-success-rate of example 7 is high enough so that 'player-1' begins executing the action specified by that example from the first iteration.

V FUTURE IMPROVEMENTS AND EXTENSIONS

Future research will extend the limits of the LRCEC system to incorporate more functions currently performed by the user. Once the extensions have been implemented the following learning issues can be addressed: (1)Determine the limits of this paradigm for learning (2)Ascertain the knowledge and proceses necessary to learn a sequence of examples (i.e. sequence of actions) (3)Evaluate accommodation techniques and (4)Compare how well the system learns as a function of the expert supplying the knowledge.

VI CONCLUSION

This paper has demonstrated how some types of learning can be accompolished by applying the CEG paradigm to examples representing domain knowledge.

ACKNOWLEDGEMENTS

Many thanks and much gratitude goes to Prof. Edwina L. Rissland for her invaluable guidance, critiques, suggestions, and research that provided the foundation for this paper; Daryl T. Lawton for smoothing out my thoughts; and to the COINS department for free computer time on weekends.

BIBLIOGRAPHY

[1] Collins, A. and A. Stevens, (1979) "GOALS AND STRATAGIES OF EFFECTIVE TEACHERS" Bolt Beranek and Newnan Inc., Cambridge, Mass.

[2] Rissland, E., and E. Soloway (1980) "OVERVIEW OF AN EXAMPLE GENERATION SYSTEM" Proceedings of First National Conference on Artifical Intelligence

C3) Rissland, E. (1980) "EXAMPLE GENERATION" Proceedings of Third National Conference of the Canadian Society for Computational Studies of Intelligence

[4] Rissland, E. L., and E. M, Soloway "CONSTRAINED EXAMPLE GENERATION: A TESTBED FOR STUDYING ISSUES IN LEARNING". In Proc IJCAI-81

[5] Rissland, E., Soloway, E., O'Connor, S., Waisbrot, S., Wall, R., Wesley, L., T. Weymouth (1980) "EXAMPLES OF EXAMPLE GENERATION USING THE CEG ARCHITECTURE" COINS Technical Report, in preperation.

[6] McDERMOTT, J. (1978) "ANA: AN ASSIMILATING AND ACCOMMODATING PRODUCTION SYSTEM" DARPA Report 3597