acm

SIGDA NEWSLETTER

SPECIAL INTEREST GROUP ON DESIGN AUTOMATION

Contents:

SPRING 1973 VOLUME 3 NUMBERS 1 and 2

SIGDA SPECIAL INTEREST GROUP ON DESIGN AUTOMATION

The goal of SIGDA is:
To enhance the utility of computers as engineering tools in the design, fabrication, and test of equipment and systems.

The objectives of SIGDA are:

1) To provide a means for the exhange of information concerning:
   a) Techniques
   b) Algorithms
   c) Computer Programs

   relevant to the computer assisted design of equipment and systems.

2) To publish quarterly newsletters containing news, technical information and opinions of interest to those involved in design and design automation.

3) To work closely with other ACM and professional committees and activities concerned with computer utilization as a design tool.

CHAIRMAN:

Charles E. Radke
IBM Corporation (B31/965)
P. O. Box 390
Poughkeepsie, New York 12602
(914) 463-4074

VICE CHAIRMAN:

Lawrence Margol
North American Rockwell
P. O. Box 3669
Anaheim, California 92803
(714) 632-8565

SECRETARY/TREASURER:

John R. Hanne
Texas Instruments (M/S 907)
P. O. Box 5012
Dallas, Texas 75222
(214) 238-3554

EDITOR:

Stephen Krosner
1581 NW 7th Street
Boca Raton, Florida 33432

MEMBERSHIP

SIGDA dues are $3.00 for ACM members and $5.00 for non-ACM members. Checks should be made payable to the ACM and may be mailed to the SIGDA Secretary/Treasurer listed above, or to SIGDA, ACM Headquarters, 1133 Avenue of Americas, New York, N.Y. 10036. Please enclose your preferred mailing address and ACM Number (if ACM member).

SIG/SIC FUNCTIONS

Information processing comprises many fields, and continually evolves new subsectors. Within ACM these receive appropriate attention through Special Interest Groups (SIGs) and Special Interest Committees (SICs) that function as centralizing bodies for those of like technical interests...arranging meetings, issuing bulletins, and acting as both repositories and clearing houses. The SIGs and SICs operate cohesively for the development and advancement of the group purposes, and optimal coordination with other activities. ACM members may, of course, join more than one special interest body. The existence of SIGs and SICs offers the individual member all the advantages of a homogeneous narrower-purpose group within a large cross-field society.

ACTIVITIES

1) Informal technical meetings at SJSS and FJCC.
2) Formal meeting during National ACM meeting + DA Workshop.
3) Joint sponsorship of annual Design Automation Workshop.
4) Quarterly newsletter.
5) Panel and/or technical sessions at other National meetings.

FIELD OF INTEREST OF SIGDA MEMBERS

Theoretic, analytic, and heuristic methods for:
   1) performing design tasks,
   2) assisting in design tasks,
   3) optimizing designs

through the use of computer techniques, algorithms and programs to:
   1) facilitate communications between designers and design tasks.
   2) provide design documentation,
   3) evaluate design through simulation,
   4) control manufacturing processes

EDITORIAL

Last issue I entered a plea for material. The response was somewhat underwhelming. I would like to believe that the reason for this is that you are all busy inventing design automation systems, but I feel that this may not be the case.

More realistically, the problem is most likely in two parts. Either you feel that what you have to say is not important and that even if it was you couldn't convince anyone else via the printed word, or your boss says that your work is of a proprietary nature and he would prefer that you do not disclose your breakthroughs.

I would like to comment on the first reason since the second one is not really open for fruitful solution.

I am sure that we all agree that any result is trivial once someone has clearly explained it.

The question you should ask yourself is, "Did I see the solution clearly when I started, or did I have some difficulty getting to it?" If you had difficulty, so will many others. Even if you didn't find it hard, how many other people have your blend of skill and background? Talk it over, write it down, and send it to me. Let me decide about publishing it.

Submitting a paper is easy once the writing is complete. Just type it single spaced on one side of size 8 1/2" by 11" paper. Number each page lightly on the back in pencil---do not type the page numbers. Drawings should be in black on white paper. Do not send copies. I will return originals if you request it. Unfortunately, our publication budget will not allow us to print halftones, so line drawings only.

We are trying to publish once a quarter in January, April, July, and October. I like to have all material in hand by the 15th of the previous month.

This issue, unfortunately, is somewhat late in coming. I have been holding it awaiting a technical paper that I hoped you would find interesting.

In December of 1972, I had the privilege of attending the Israel Scientific Research Conference on Design Automation. One of the comments made consistantly by the Israeli attendees was, "Can you prove that? Is it algorithmic?" We answered as you might have guessed, "No, but it works on 9n% of the cases we have tried."

I returned from Israel feeling that a case for heuristic solutions to problems should have been presented at the conference. I was pleased to find in an AIIE publication just such a case, and I obtained permission to reprint it for your information. I hope you will find it as useful as I have.

S. P. Krosner

Chairman's Message-C. E. Radke

Elections

You will have received an election notice by the time you receive this issue of the newsletter. The last election was held in the fall of 1970 for the term 1971 to 1972.

The slate of officers is a good one and I wish to thank Dr. John Hanne for heading up the nominating committee along with John Rini and Don Humeke as members. The slate is as follows:

Chairman:	Charles E. Radke IBM Poughkeepsie, New York
	Stephen Krosner IBM Boca Raton, Florida
Vice Chairman:	Lawrence Margol North American Rockwell Anaheim, California
	David Hightower Bell Telephone Laboratory Holmdel, New Jersey
Secretary-Treasurer:	Stephen Szygenda Southern Methodist University Dallas, Texas
	Loma Capodanno Bell Telephone Laboratory Whippany, New Jersey

Members should remember that candidates can be nominated by petition from the members at large. Our by-laws read that signatures of ten members are required. The by-laws also require at least two nominees per office.

This time around I welcome my running mate (or really my opponent) Steve Krosner. I say running mate because I feel our objectives are very similar. Steve has been very active in SIGDA and as you know presently has the position of SIGDA Newsletter Editor.

May the man with the most votes win!

1973 DA Workshop

The new chairman will take office at the 1973 DA Workshop in Portland, Oregon during June 25-27, 1973. That is not the only reason for you to attend. I encourage you to attend first, because it is our workshop, a three day set of technical sessions devoted strictly to Design Automation. Second, these sessions provide you with the latest in techniques, concepts, and systems for DA. As a member of SIGDA you will automatically receive the announcement and agenda of the workshop. Plan now to attend. Steve Krosner tells me salmon fishing is legitimate during the time of the workshop. So before or after (not during) the workshop you can take a few days off and enjoy yourselves. We'll see you there.

ACM '73

This year as in the past two years, SIGDA will have two sessions at the National ACM Conference. The plans include a paper session chaired by Chuck Radke and interface session with SIGRAPH depicting the difference between the systems designers' view of an interactive CAD/CAM System and the users' view. The second session will be chaired by Steve Krosner.

I hope to see you all in Atlanta August 27 - 29.

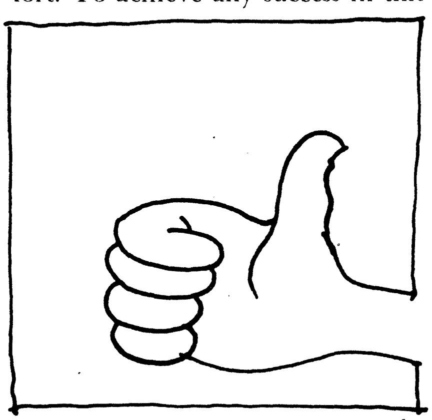
Heuristic programming solution techniques are proposed as a panacea for problem-solving by some, condemned by others. This article presents a philosophy for determining whether or not to use a heuristic program, and, if so, selecting a proper heuristic program, and validating its performance.

J. P. Ignizio, R. M. Wyskida, M. R. Wilhelm, University of Alabama in Huntsville, Huntsville, Alabama

A rationale for heuristic program selection and evaluation

In the past, great claims have been set forth as to the potential of the heuristic procedure or "heuristic programming," References 1 and 2. As early as 1958 Simon and Newell, Reference 2, wrote of "Heuristic Problem Solving: The Next Advance in Operations Research." However, and this is strictly the opinion of the authors (although it is shared by most others we have discussed the matter with), rather than seeing a relative increase in heuristic program solution techniques in the literature, the converse appears to be true. Most of the journals dealing with operations research, management science, industrial engineering, and related fields have evolved (or perhaps dissolved, depending on one's point of view) into journals of mathematics, or perhaps abstract mathematics. The fundamental idea of engineering as being directed toward real world applications often seems to have been lost. As a result, we (industrial engineers/operations researchers) are now caustically described as pseudo-mathematicians, and the field has been defined as the "science of providing elegant solutions to trivial problems." OR has even been credited with such things as our poor performance and defeats in the Indo-China war, Reference 5, cost overruns on defense contracts and so on. Such criticism comes from both outside and within our profession, References 3, 4, 5.

Even under this barrage of criticism, there is still a continuous stream of "elegant solutions to trivial problems" being turned out while real world problems are not-so-deftly side-stepped. It is recognized that the more rigorous mathematically inclined research is an absolute necessity. However, it is also our duty to concentrate a more balanced effort toward the immediate solutions of real problems of real sizes. Much, much more attention needs to be directed toward this latter effort. To achieve any success in this field, however, more use must be made of the heuristic method of problem solving.



Unfortunately, heuristic problem-solving is rarely taught or even discussed in our universities and, quite often, the first encounter that the graduate engineer has with a non-trivial problem occurs after his graduation. To alleviate this situation, a straightforward philosophy of heuristic program selection and validation is needed. Such a philosophy could easily be included in a senior level seminar whose purpose would be to prepare the student for a more successful and easier entry into the industrial climate.

When use a heuristic program?

The contention of the authors is that the heuristic method forms a vital, but neglected, part of the operations research analyst's tools. It is not suggested, however, that every problem should be solved by heuristic programming. In fact heuristic programming should only be considered if it is obvious that all other methods would fail (however, this usually does occur when we discuss real world problems). The point is that heuristic programming should only be used where exact techniques are not available and/or not economical.

The two principal uses of the heuristic programming approach are:

1. to solve problems of such size that exact and/or more elegant methods cannot be employed.
2. to obtain a good, if not optimal, starting point for the more elegant procedures.

What is a heuristic?

Simon, Reference 6, refers to heuristics as rules of thumb selected on the basis that they will aid in problem solving or contribute to a reduction in the average time spent in searching for a solution. Making use of these definitions, a heuristic

Reprinted from Industrial Engineering, January 1972.

program is defined herein as a problem-solving program, consisting of a combination of heuristics, explicitly stated, which utilizes certain common-sense principles or devices to derive *acceptable,* although not necessarily optimal solutions.

Heuristic programming techniques are most often used when the problem is of such size and complexity that exact optimizing algorithms are not available or not practical. The heuristic programming approach seeks solutions based upon acceptability characteristics rather than optimizing rules. It gives explicit consideration to a number of factors (for example, computer storage capacity and solution time) in addition to the quality of the solution produced. Further, the evaluation of heuristic programming techniques is usually done by inductive rather than deductive procedures. Specific heuristic programs are justified, not because they attain an analytically verifiable optimum solution, but, rather because experimentation has proven that they are useful in practice.

Two of the most difficult aspects of dealing with heuristic programming techniques are in determining the heuristic program to use, and in validating the performance of the heuristic program. We first discuss the process of selecting the heuristic program.

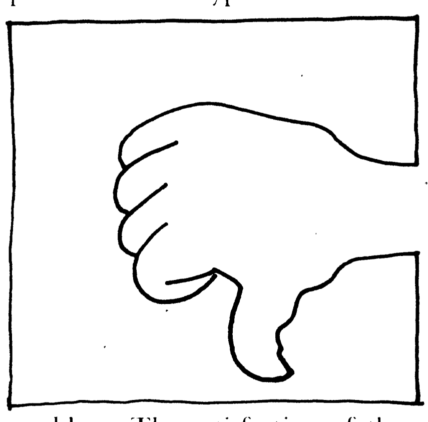
Selection philosophy

Essentially there are an infinite number of heuristic programs to choose from. Thus, for any heuristic program selected it is quite easy for the critic to challenge its selection versus other possible choices. Since it is not feasible to evaluate all possible heuristic programs, some other more practical selection procedure is necessary.

It is advocated here that the method of selection used be based on a pre-established "aspiration level." The aspiration level should indicate the aspired level of all pertinent attributes such as problem size limits, computation time, economic factors, desires of the decision-maker, and so forth. After setting these levels, heuristic rules are generated (intuitively), beginning with the most simple. Once a combination of such rules (or heuristic program) is devised that does satisfy the levels of aspiration, the selection process ends. This approach is the method actually used for most complex real world problems. For example, in hiring a new employee we establish certain criteria for employment and usually hire the first applicant satisfying these criteria. Simon and March, Reference 7, advocate the following hypothesis: "Most decision making, whether organizational or individual, is concerned with the discovery and selection of satisfactory alternatives; only in exceptional cases is it concerned with the discovery and selection of optimal alternatives." This hypothesis suggests that in most cases the search for alternatives will not be exhaustive. Search is continued only until an alternative (heuristic program) is found which is *acceptable.* Such a philosophy seems the only practical one for the selection of heuristics and, in addition, establishes an orderly, systematic procedure for selection.

Aspiration levels

The aspiration levels for the heuristic program under consideration should be established after a survey of existing solution techniques and with consideration given to what attributes would most likely be pertinent in a typical real world



problem. The satisfaction of these aspiration levels serve as a measure of the heuristic program's performance.

Often, in the literature, one is led to believe that the time attribute is the one essential factor upon which selection or comparisons of methods should be made. For example, publications often cite how much faster a specific problem or problems can be solved versus some other method. Unfortunately, such appraisals are misleading at best and worthless at worst. The problems used for comparison are usually very small problems relative to those actually encountered in practice. What difference then, does it make if one method is "faster" than another if neither can solve real world problems?

Consequently, rather than stressing speed on small problems, the choice of heuristic program should be based on much more important factors such as time requirement predictability, the time required to solve very large problems, and the size versus time characteristics of the method.

Validating the heuristic

In a previous section it was stated that "specific heuristic programs are justified, not because they attain an analytically verifiable optimum solution, but, rather because experimentation has proven that they are useful in practice." Since experimentation on the actual real world problem is usually impractical, other, more feasible, testing methods must be used. Possible alternatives include:

1. Comparison of the heuristic program results with those of exact methods.
2. Solving problems with known solutions (problems previously solved by exact methods).
3. Generation of large problems with known solutions to be used as test cases.

Methods (1) and (2) are those found in most of the publications the authors have encountered. Some argue that these two methods cannot justifiably be used to either validate or reject a heuristic scheme. The reason for their conclusion is as follows. The heuristic is usually, if not always, intended for the solution of problems of real world (i.e., large) size whereas exact methods are almost invariably restricted to unrealistically small problems. Such a comparison then makes no more sense than to say, compare the results of hammering in a nail with a regular hammer versus using a sledge hammer. The sledge hammer is a crude, inefficient (but undeniably fast) way to hammer small nails but quite efficient in dealing with the tasks it is really intended for, driving in spikes or stakes. They then conclude that it makes no more sense to evaluate the performance of a heuristic, designed for problems of thousands of variables, against an exact method good only for problems of a few dozen variables.

Consequently, they advocate that the heuristic can only be judged fairly by performing this judgment on its effectiveness in solving real world size problems. As one means to accomplish this comparison, it is sometimes possible that problems of large sizes can be generated with known exact solutions. R. Roth, Reference 8, describes one technique to accomplish this and Ignizio and Case, Reference 9, present another, for the problem of validating heuristics for solving the set-covering problem.

The authors agree, in principal, with such reasoning. However, it is not always possible to generate such large problems with known solutions. Thus, one has to either base his confidence in a heuristic program by either method (1) or (2) as listed above, or else give up the idea of using a heuristic program. The latter alternative, however, is certainly unwise if we are dealing with a real world problem which must be solved. It is advocated that we base our acceptance of the heuristic then, on "faith" where such faith is developed when we are satisfied with the "reasonableness" of the heuristic program. This faith is then intensified if the heuristic program results agree (to some aspired degree) with those of exact methods on small problems and/or if the results of the heuristic are satisfactory on small problems which have been previously solved. Such a judgement is, of course, bound to be condemned as subjective and unscientific. However, we simply cannot wait until the theorists have developed a truly exact method and, as engineers, must proceed with the job of solving the problem with the resources at hand.

Example

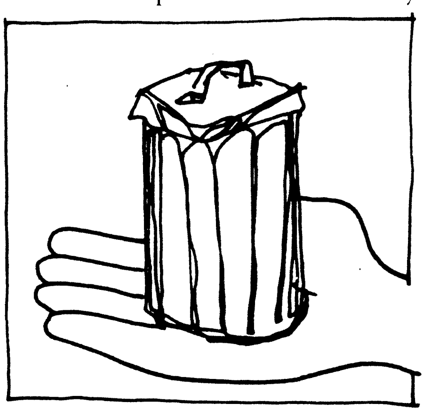
Heuristic programming is admittedly an art which, although its general philosophy may be taught, is best learned through examples and actual experience. To illustrate the philosophy of this paper as put into actual practice, a discussion of the solution of an actual real world problem will be utilized.

The city of Huntsville, Alabama, contracted (1970-71) with the University of Alabama in Huntsville to establish improved garbage collection routes for the entire city, Reference 11. The aspirations of the city were to: (1) reduce the total number of collection routes from 48 to 42, (2) establish more equitable routes (previous collection routes varied substantially in size and the collection time required), (3) keep the collections on the two-day (either Monday/Thursday or Tuesday/Friday) basis, and (4) provide an incentive to the workers by sizing the individual collection routes on a heavy volume day basis (thus on light collection days the workers could possibly go home early).

The limited amount of time and funds (September 1970 — February 1971 and $13,401, respectively) made it necessary that a collection routing scheme be established to satisfy the city via relatively unsophisticated means. However, this lack of sophistication is actually felt to be a major advantage of the scheme since it has been observed that so-called "optimal" and more elegant methods lack the flexibility of the heuristic programming method employed.

The heuristic programming approach utilized to solve this problem was developed as follows:

1. The aspirations of the city



were defined (as noted above) and used as measures of the heuristic program's performance.

2. The previously existing collection routing scheme was investigated to establish if it could be modified or whether an entirely new scheme was required. The previously existing scheme appeared to have "grown like Topsy" over the years and was a collection of fragmented routes of widely varying collection times. This fragmentation, size discrepancies, and lack of accurate collection time data precluded any minor modifications and thus an entirely new scheme was developed.

3. Heuristic rules, upon which the heuristic program was to be based, were formulated. These rules were:

a. A collection route will not be fragmented (i.e., each route will be a continuous route and will not be broken into scattered sections).

b. Each individual route would be compact or "clustered" rather than spread out (i.e., it would be attempted to minimize the maximum distance from any two points within the route).

c. Major thoroughfares, man-made boundaries (railroads, drainage canals, etc.) and natural boundaries (mountains, rivers, etc.) provide the necessary constraints when seeking to achieve step b above.

d. Contiguous streets have preference over streets which tend to leave the general location of the continuous path.

e. Truck path decisions at intersections would be based on the desire to satisfy the four heuristic rules directly above.

f. Route collection times would be based on the heavy collection day load (data for these times was obtained through a work sampling program implemented by the University). Consequently, the workers should be able to complete the assigned route early on light collection days.

g. Each route was to take more than 7½ hours but less than 8 hours (again, based on heavy collection day data) including driving time to and from the route and unloading of the truck.

The collection of heuristic rules (a) through (g) form the basis of the heuristic program developed.

4. A single, continuous route was developed for traveling through every street in the city. This route was then segmented into subroutes which represented individual collection routes. A collection route was obtained as soon as the sum of collection, unloading, and driving time was between 7½ and 8 hours.

Step (4) above needs to be considered in more detail since it represents the actual implementation of the heuristic program. The single, continous route developed in step 4 could have been solved via computerized routing algorithms. However, it was decided to use human analysts for this step. These analysts traced a single, continuous, "common-sense" route through the entire city. It was, and is, firmly believed that solving such a complex pattern search problem can be accomplished better by a human than a computer. There is simply no way to transfer to the computer, the lessons learned by the experience of driving a garbage truck through a metropolitan area. Consequently, the single, continuous route through the entire city was drawn, using the knowledge of three analysts, by hand on a city map. Decisions as to turns at intersections, backing through alleys, encounter of cul-de-sacs, dead-end streets and the like were based on the desire to satisfy the heuristic rules stated above and to provide a reasonable and practical driving pattern for the garbage trucks.

This single continuous route was then broken down into individual collection routes whose length was based on an estimated collection time of between 7½ and 8 hours (including driving time to and from the route).

The routes decided upon were then provided to the city. The scheme has now been in use since the summer of 1971 and has proved to be successful, the ultimate verification of a heuristic program.

References

(1) Tonge, F. M., "The Use of Heuristic Programming in Management Science," *Management Science,* Volume 7, April, 1961.

(2) Simon, Herbert A. and Allen Newell, "Heuristic Problem Solving: The Next Advance in Operations Research," *Operations Research,* January-February, 1958.

(3) Caywood, Thomas E., "How Can We Improve Operations Research?", *Operations Research,* July-August, 1970.

(4) Spitzer, Murray, "The Computer Art of Schedule-Making," *Datamation,* April, 1969.

(5) Anderson, Jack, "Worse than the Viet Cong — The Paper Clip War," *Parade,* December 8, 1968.

(6) Simon, H. A., "Modeling Human Mental Processes," Proceedings of the 1961 Western Joint Computer Conference.

(7) March, J. G., and Herbert A. Simon, *Organizations,* New York: John Wiley and Sons, Sons, Inc., 1958.

(8) Roth, R., "Computer Solutions to Minimum-Cover Problems," *Operations Research,* June, 1969.

(9) Ignizio, James P. and Kenneth E. Case, "A Method for Validating Certain Set-Covering Heuristics," in review.

(10) Ignizio, James P., "A Heuristic Solution to Generalized Covering Problems, unpublished Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1971.

(11) Wyskida, R. M., Wilhelm, M. R., and Mercier, J. L., "Refuse Collection Operations Study," UARI Research Report No. 103, Research Institute, University of Alabama in Huntsville, February, 1971.

James P. Ignizio is a Senior Research Engineer at the University of Alabama in Huntsville. Previously, he was associated with the aerospace industry. He holds a BS in electrical engineering from the University of Akron, an MS in engineering from the University of Akron, an MS in engineering from the University of Alabama in Huntsville, and a PhD in industrial engineering and operations research from Virginia Polytechnic Institute and State University. A senior member of AIIE, Dr. Ignizio is also a member of TIMS, ORSA and SCI.

Richard M. Wyskida is Chairman, Industrial and Systems Engineering, the University of Alabama in Huntsville. Before joining the University, he was in research with the Marshall Space Flight Center. He has a BS in electrical engineering from Tri-State College, an MSIE from the University of Alabama, and a PhD in IE from Oklahoma State University. Dr. Wyskida is a registered professional engineer, a senior member of AIIE, and a member of ORSA, ASEE, and Alpha Pi Mu.

Mickey R. Wilhelm is completing the requirements for a PhD in industrial and systems engineering at the University of Alabama in Huntsville, where he also serves as a graduate assistant.

At Dr. Perlin's top-them-all ACM 73.

Come on down to Atlanta for ACM 73 August 27-29. You'll find it's quite a city—one of the most exciting in the country. With tomorrow's ideas and yesterday's charm. Top dining spots. Underground Atlanta. And outstanding landmarks. Like the Hyatt Regency Atlanta—with its 22-story high lobby—scene of what should be the best ACM annual conference of them all.

Dr. Irwin Perlin of Georgia Tech is ACM 73 chairman. He's chosen the theme "Computers in the Service of Man," to focus on how computers are promising and producing significant benefits to society. It's a fitting theme to help launch ACM's second quarter century—with ACM's Special Interest Groups contributing heavily to the technical program.

"It's especially significant to have this conference in Atlanta," says Dr. Perlin. "Some of the most advanced and sophisticated applications of computers are to be found here in The New South. And some of the most beneficial uses. ACM 73 should bring out a lot of new ideas."

You'll be in good company at ACM 73. More and more of our 29,000 computer professionals are attending our annual conferences. If you're already an ACM member, plan to attend. If you're not, join us there and convert part of your admission fee to annual dues. Send in the coupon today!

Association for Computing Machinery
1133 Avenue of the Americas
New York, New York 10036

I would like to consider ACM and ACM 73. Please send more information.

Name ______________________

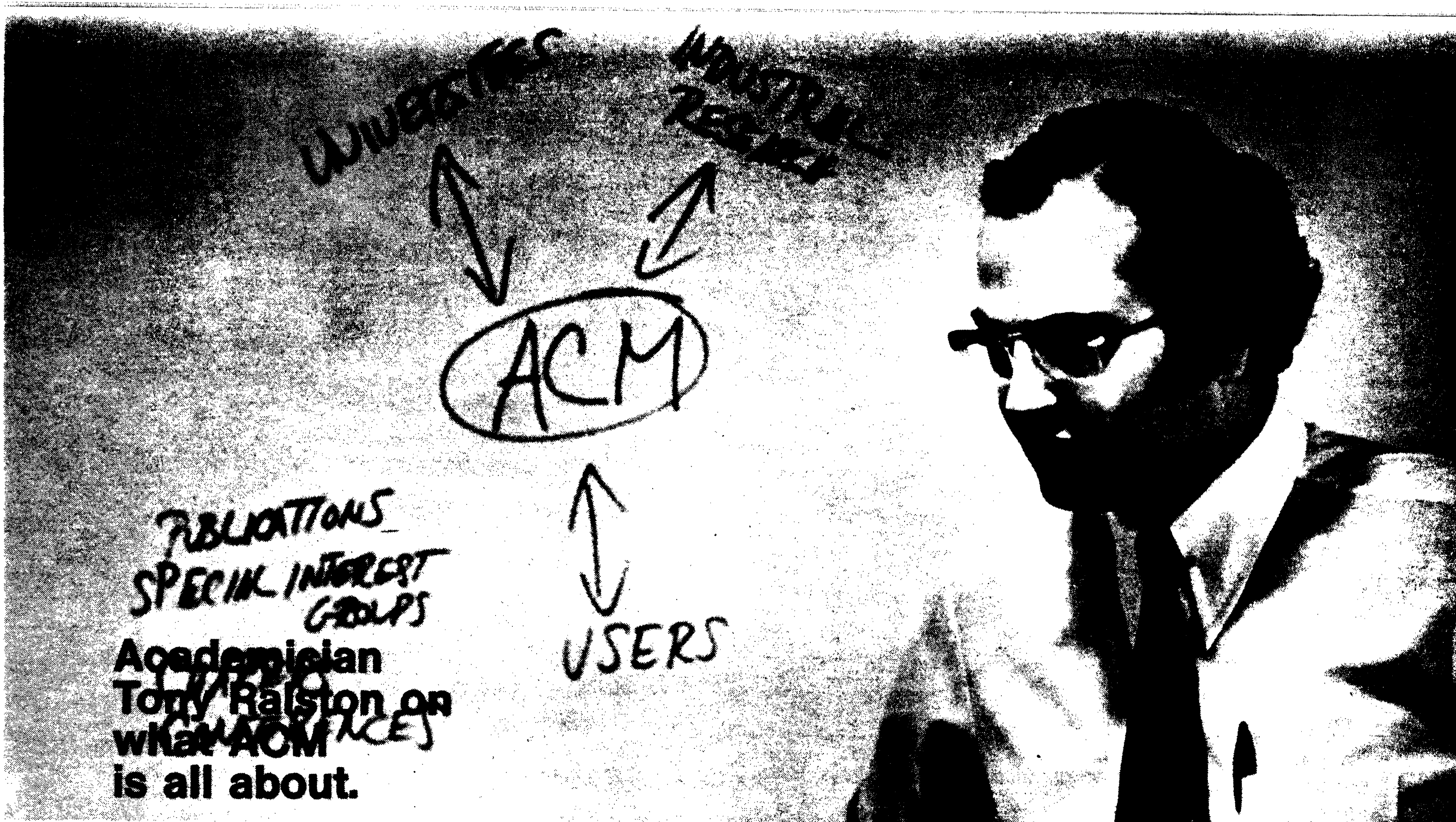
Position ______________________

Address ______________________

City __________ State __________ Zip __________

Association for Computing Machinery

DA 6/73



Academician Tony Ralston on what ACM is all about.

A three-way bridge to technical vitality.

Tony Ralston is the new President of ACM. He's a colorful professional and academician with a long career in numerical analysis and computer science. He has some strong ideas on the goals of ACM and the needs it fulfills for computer professionals.

"ACM was once an academic society," says Tony. "But now we're far more than that. We represent a sort of three-way bridge between the academician, the research and development scientist and the practitioner. The person at the leading edge of computer science, the individual in the hardware and software lab and the user applying computers to practical business and scientific problems. We try to stimulate cross-talk between all three.

"Getting research results into the mainstream of computing isn't easy. It needs this kind of interaction. Active participation. A professional commitment to both the science and the practice of computing. That's when computing becomes a career—and not just another job. That's technical vitality. And that's what ACM is all about."

ACM chapters, special interest groups, publications and conferences are all examples of this kind of dialogue. Along with seminars, lectures and many other activities. Maintain your technical vitality and make a contribution to the computer profession. Join ACM. Send in the coupon today!

Association for Computing Machinery
1133 Avenue of the Americas
New York, New York 10036

I would like to consider joining ACM. Please send more information.

Name ________________

Position ________________

Address ________________

City ______ State ______ Zip ______

acm

Association for Computing Machinery

DA 6/73

Association for Computing Machinery
1133 Avenue of the Americas, New York, New York 10036

Non-Profit Org.
U. S. POSTAGE
PAID
New York, N. Y.
Permit No. 2397

N095422 07 SDA
CHARLES E RADKE 7319
IBM CORP
DEPT B19 BLDG 965
PO BOX 390
POUGHKEEPSIE NY 12602