Communication and Control in Man and Machine Translation

DAVID G. HAYS

Metagram and Planning Research
Corporation International,
Hamburg, New York, USA

Any system operating with input data of various kinds must be self-regulating in order to be effective. The author gives arguments concerning complex systems in general as well as MT systems, operating with data from cybernetics, psychology, computation, and linguistics. General perspectives are reviewed for constructing a self-regulating and improvable MT system.

Between 1957 and 1960, The RAND corporation issued a series of Research Memoranda in which I developed, with the collaboration of several programmers, Slavicists, technologists, and aides, a scheme for machine translation. The first to appear was a *Manual for Keypunching Russian Scientific Text.* The last was *Russian Sentence-structure Determination.* It was dated April 1, 1960; but the idea that it developed in detail had been stated in *The Use of Machines in the Construction of a Grammar and Computer Program for Structural Analysis,* dated January 9, 1959. (The latter paper, and its title, combined two proposed contributions to the UNESCO conference in Paris that led to the foundation of the International Federation of Information Processing Societies. We — Kenneth E. Harper and I—had to be content with one spot on the program.) Returning to machine translation after an interval of eighteen years, I submit a reasoned defense of an approach which, when I invented it, derived from intuition only. The argument draws on cybernetics, computation, and linguistics.

CYBERNETICS

The cybernetic theory of Norbert Wiener describes a process of regulation by gradual reduction of error. In my childhood I learned of the regulator or governor used on the steam engine. The engine uses a tiny fraction of its power to spin an axle. A rod-and-ball hangs from the axle by a pivot that allows it to move further outward as the axle spins faster and faster. By some arrangement of cams and linkages, the angle of the rod alters the opening of the valve that passes steam from boiler to engine. If the axle spins fast, the ball swings out and the valve closes a little. If the axle spins slow, the ball swings in and the valve opens a bit. This nineteenth-century invention belongs to the class of regulators that Wiener described in a coherent mathematical theory.

The importance of cybernetic regulators grows out of the unpredictability of nature. The load on the steam engine can be small or large; the amount of steam to produce constant speed varies from moment to moment; the governor computes the appropriate steam supply as a function of two quantities: intended speed and actual speed. The flow is a function of the difference.

The intended speed is a goal. The actual speed is an input. The valve setting is an output. The difference between input and goal is an error. The governor of a steam engine has no method of considering all possible solutions of its problem. The governor does not test first one solution and then another. Its computation takes time, as all real processes do. During the interval of computation, the error gradually diminishes.

In 1973. following a long period of private investigation, William T. Powers published *Behavior: The Control of Perception.* In this book he describes the activities of animals in their environments, including in his broad sweep the activities of human beings in diverse situations. He uses the cybernetic mechanism of Wiener, but combines several such mechanisms in a stack: The output of one mechanism sets the goal for another. Input from the environment to the organism is not a stimulus or trigger to a response; instead, it serves the organism as the spin rate of the axle serves the governor. By comparison of input and goal, the system can compute an error quantity. The input can be described, in the term used by the historian Arnold Toynbee, as a challenge. Powers discusses many questions in his book: The interaction of genetics and experience; the mechanisms of reorganisation that alter connections between challenge and response during the lifetime of an orga-nism, and others. After a few years of silent contemplation, both the academic psychologists and the popularizers of computing have begun to give attention to Powers's theories.

COMPUTATION

I have often cited John von Neumann as the inventor of the computer, and I think that I am both correct and in accord with the opinions of many if not all others on this question. In the current literature, at least in the United States, the von Neumann machine is considered obsolete; but I believe that von Neumann's idea was so abstract that many realisations of it are possible, and therefore that only one realisation — adopted by almost everyone for the last thirty years — is obsolete. To the theory of computation John McCarthy contributed a fundamental distinction, the distinction between the formulation of a solution and the construction of a path to solution. McCarthy's programming language LISP appears in many places; but his fundamental idea does not. That the idea is important can be seen from the simple formulations of the solutions to many complex problems. If the problem is to place eight queens on a chessboard in such positions that they do not attack each other, then the form of the solution is 'eight queens on the chessboard that do not attack each other'. In the formulation of the solution, the programmer cares neither when nor where calculation and storage take place; in the original version of LISP, the programmer could not specify time or place of calculation. At present, it is commonplace to disregard the physical locations for storage of data; one speaks of 'virtual' memory. It is not at all common even today to disregard the physical time of calculation, yet to make progress it seems that one must.

In a cybernetic theory of computation, the path to solution is defined in an error space. As computational time passes, the error must decrease. The applications programmer supplies the form of the required solution and an initial approximation. The systems programmer supplies a mechanism for the steady improvement of approximation. (In the original LISP, this distinction is clear.) Powers's work clarifies the distinction between object time and computational time. The user can specify a target for the arrow or the trajectory of the arrow. If the target, then the program computes an aim; but if a trajectory, then the program computes course corrections during the arrow's flight.

LINGUISTICS

Noam Chomsky introduced a distinction and supplied a pair of terms that have vexed many commentators for years. The generation of a language is different, he wrote, from the production or recognition of sentences. Despite repeated protests in Chomsky's writings, many investigators have taken the form and the sequence of rules in a transformational grammar as the form and the sequence of instructions in a psychological or computational processor. A most significant advance occurs in the work of Theo Vennemann and others who call themselves natural generativists: As in the original LISP, they admit of no sequencing of rules in grammars.

A generative grammar is the form of solution of the problem 'What is English?' or 'What is Russian?'. That a sentence consists of a noun phrase and a verb phrase is a fact of the same kind as the fact that eight queens on a chessboard not attacking one another solves the old problem. Thus the linguist has the point of view of the applications programmer.

The psychologist and the informatician must adopt the point of view of the systems programmer. Among the various theories of computation, only the cybernetic theory has given any hint of success when confronted with problems of the complexity that linguists and psychologists confront. If the psychologist adopts the cybernetic theory of computation, then the program for production or recognition of a sentence is a mechanism for reduction of error.

For the linguist, the utterance is a trajectory in object time: A movement from first syllable of recorded speech to utterance-final silence. For the psychologist or informatician, error reduction is a trajectory in computational time: A movement from first approximation to final acceptance of some interpretation of the sentence, if the sentence is given to be understood, or from first approximation to final utterance of some form of expression, if the understanding — the idea — is given to be formulated in overt language. A sentence can be taken as the farm of solution to one problem for the hearer and to another problem for the speaker. The speaker's problem is, 'How can I tell this hearer that P?' and the hearer's problem is, 'How can I understand the P of this speaker?'

The parse of a sentence can be taken as the form of solution of the problem, 'How does this sentence fit the grammar of its language?’ The solution form can be identified with the goal of a cybernetic process; the goal is a trajectory, not a target. The parse is the link between computational time and utterance time, the latter flowing from the first syllable to the last and the former moving from large error to small. Error is large if form and content are very different; error is small if form and content are very similar.

The grammar regulates processing. It is more than the form of solution to the problem 'What is English?' It is also the form of solution to the problem 'How can I find a parse of a sentence of English?' Or, to put the entire complexity into one statement, 'How can I find out how this sentence fits the form of solution of the problem 'How can I find out how this sentence fits the form of solution of the problem...?'?' The grammar is both the solution of the problem of linguistic form, and the solution of the problem of applying the grammar to an object; and since that statement is recursive in form, the grammar is the closed representation of an infinite regress.

A trajectory has a terminus *a quo* and a terminus *ad quem.* These points are fixed prior to utterance. The terminus *a quo,* or starting point, is established by the social and cultural context and by the discourse up to the moment; but the terminus *ad quem,* or target, is likewise established by the same factors. Thus the present paper is required to exhibit a trajectory toward the target of MT, and its first paragraph more narrowly defines its target. Each successive fragment of an utterance must lie within the target area as specified theretofore and add to the specification of the target area. Whatever satisfies those two conditions is a course correction.

In *Chimera,* John Barth writes that 'The key to the treasure is the treasure'. In language, the form of solution to the problem 'What is a sentence?' is the path to the solution of the problem 'What is the speaker telling me?' If the form and the path are indistinguishable, the system is selfregulating and requires no independent systems programmer to devise approximative mechanisms. Since children learn to speak before they learn to solve problems, indeed since children learn to think before they are taught to think, it seems clear that all thought is selfregulating.

Thus Chomsky's distinction between generation and production, ill-comprehended as it has been, is just as wrong as its critics have taken it to be. Jakobson's emphasis on the metalingual function of language is, on the contrary, much more profoundly correct than even Jakobson seems to know. Language ('with a capital L') is the closure of an infinite regress of metalingual jumps.

To adopt Chomsky's idea of generative grammar, and with it the distinction between competence and performance that he correctly derives from the basic idea, is to move inevitably into the cul de sac of artificial intelligence: To provide powerful methods of approximation in lieu of precise formulation of the problem. In the present context, to arrive at a good machine (although surely never a great machine) without ever approaching an understanding of the man. Computational linguistics might offer to deal with language by emulation: To provide methods for the realisation of grammars (forms of solution) in hardware such that the problem of utterance can solve itself.

THE USE OF MACHINES IN THE CONSTRUCTION OF A GRAMMAR AND COMPUTER PROGRAM FOR STRUCTURAL ANALYSIS

I end where I began; I return to the ideas of 1959 with a point of view that has changed under the stimulation of many excellent publications I have read. When I published my first remarks on grammar and parsing I was a child, and I did not understand what I was saying. I described a parser that worked on surface structure and aimed at a correct parse with no systematic treatment of ambiguities.

Emonds' suggestion that structure is preserved under transformation now makes acceptable again the old position, that surface structure is parsable. Recent findings concerning the order of adjectives in the English NP (work that Richard Fritzson should publish soon) greatly enhance the case for adequacy of linguistic constraints to quasideterministic parsing. Fritzson's present work on multistratal parsing yields a simple control structure on each stratum, suitable to the view of McCawley and others — the view that syntax is ultimately trivial.

Taken all in all, the theory of cybernetic computation and the modern evidence about language suggest that with adequate linguistic investigation the combinatorial explosion is contained and that the task of understanding spoken input can be handled with machines of limited power.

There remains only the question of the use of machines — and of men — in the construction of grammars. How do the person and the artifact work together toward the immediate delivery of a translation and the ultimate delivery of a better system for future translations? William L. Benzon and Richard Fritzson have recently coined the term 'humanly regulated translation'. The person regulates the work of the machine as the grammar regulates parsing: the person's goal is 'fully automatic high-quality machine translation', and a good translation of the present text is assurance that the system is close to its intended trajectory.

The person does not issue instructions to the machine, since that would make the person a programmer, and translators are neither willing nor able to become good programmers. The machine does not issue instructions to the person, since the person's capacity is inherently too complex for the machine to instruct (programming human beings is not a job for any machine). Regulation of complex systems is very poorly understood; the point is to see errors, to localize them, to see the level of organisation of the machine at which the fault lies, and to point in the direction of success.

In 1959 I wrote of a computer program that I could demonstrate. In 1979 I write of a new program that I hope to be able to demonstrate. Thus I seem to have dropped to a lower level of activity; but I hope that I have been moved to a higher level of understanding by reading the work of many vigorous contributors during these twenty years.