Skill saturation: Rationalization and post-industrial work

A. ANEESH
Rutgers University, aneesh@rci.rutgers.edu

The last few decades in the United States have witnessed an exponential rise of information technologies in production and work. Advanced information systems have not only been adopted by all sectors of the economy in terms of usage and investment;[1] they have also triggered a qualitative shift in skill requirements and work experience. This study focuses on the changing structures of work skills, seeking to develop a normative hinge for a better understanding and evaluation of skills. The term "information technologies" may refer to a number of developments such as telephones, computers, microprocessor-based products, integrated telecommunications, and even to genetic engineering.[2] As my intent in this article is to explore work skills, I limit its usage to imply technologies that require the worker to interact chiefly with electronic text and graphics, whether to control heavy machines or compose a business document.

Harry Braverman[3] was one of the first to analyze the dynamics of skill transformation, generating considerable debate and research in sociology. His central concern that industrial capitalism harbors a predominant tendency to degrade workers' skills by breaking down complex tasks into simple mechanical steps has been both supported[4–8] and opposed[9,10] by subsequent research. The ensuing stalemate has forestalled a serious re-examination or resolution of the question of skill. In this article, I attempt a reconceptualization of the actual and possible degradation of skills. Keeping in view the transformation of work and development of information technologies since Braverman, I analyze the problem of skill not in terms of task splitting, but of closure of spaces for play in the structure of skills, even if tasks remain relatively integrated. I term this closure of spaces as "skill saturation," a phenomenon characterized by an absolute predictability of procedure and outcome, resulting from an exhaustive ordering of various

Theory and Society **30**: 363–396, 2001.

components of skills, and the elimination of all irregular spaces of work. Although the thesis of skill saturation is intended to be valid in both industrial and post-industrial contexts, the major part of the discussion is devoted to post-industrial work.

Eschewing a possible allusion to any progressive decline in general skill level, I argue that processes of skill saturation are neither linear nor ubiquitous. Yet, we do need analytical tools to think and talk about actual and possible problems regarding skills. The concept of skill saturation, as presented later, provides evaluative criteria by determining the extent to which the new skills have spaces for play embedded in them. The structure of this article is as follows: first, I summarize various debates on the problem of skills; second, I situate skill formation within the framework of rationalization, elucidating the shift from industrial to information work. Then I discuss the thesis of skill saturation, with specific reference to information technologies and the history of programming languages. Later, I discuss two phenomena related to new skills that this study identifies – first, the paradoxical phenomenon of workers experiencing more task control as well as intensification of work, and second, the growing homogeneity of skills across different occupations. Lastly, I discuss possibilities of resistance to skill saturation in skill, non-skill, and sociopolitical contexts.

Skill debates

Braverman[3] attributes the problem of skills to what he terms "the manufacturing or detailed division of labor" – a product of capitalism – distinguishing it from "the social division of labor" that has been prevalent in all times. According to him, capital logic splits complex tasks into simple repetitive steps in order to hire cheap unskilled labor to do the resulting detail and specialized work. To enhance its control over workers and to reduce its dependence upon them, management seizes control over the production process by acquiring the abstract/conceptual knowledge of production, reducing work performance to a mechanical execution of simple tasks. Many scholars, however, claim that the impact of technology on the content of work is more variable, lowering requisite skills in some respects but upgrading others.[9, 11–14] Others have contended that Braverman ignored the influence of labor's resistance against managerial attempts to simplify the labor process. De-skilling loses its status as an unhindered process or a determinate result of capital logic if we take into account the opposition from

labor.[15–17] Conversely, Michael Yates[18] argues that Braverman – far from ignoring workers' resistance – provides an invaluable weapon in that resistance.

With the development of information technologies, however, the question of skill assumes different dimensions. Information technologies that deal chiefly with "bits" (information) as opposed to "atoms" (material)[19] are seen by some to have created pressure for a profound *re-skilling*. Hirschhorn[20] contends that the problem of skilled machines and de-skilled workers that haunted the Industrial age has been resolved through computer integrated work, as new skills seem to combine many different functions in a complex unit of production. Despite differences across industries in the specific forms of computer technology, some scholars believe that computer workers increasingly have become more trained than in the past.[20, 21] Braverman's thesis is similarly questioned by some other scholars, who noticed a marked change in skill requirements for computer-mediated work. Although computers displaced the mechanical and most reductionist labor, those workers who survived the displacement in automated plants experienced the process of re-skilling, multi-skilling, and up-skilling, as they were retrained with a broader skills base.[21–23]

In this context, Cynthia Cockburn's[24] suggestion – to unpack the notion of "de-skilling" by separating the idea of "loss of skill" from "degradation of work," "deskilling of a group" from "deskilling of an enterprise," and "loss of skill" from "loss of control" – is significant. In fact, I point out later how skill saturation may result in greater control over the task at hand. It must be noted that Braverman did not himself use the term "de-skilling," which has come to be used to summarize his thesis of work degradation. Although some scholars have good reason to argue that information technologies have overcome the problem of what Braverman called the detailed division of labor, many others believe that the theoretical possibilities of Braverman's discussion of skills are yet to be exhausted.[18, 25–27] His distinction between "skill" and "dexterity," his emphasis on the time factor for learning different skills, and his overall concern about work degradation still seem relevant issues. Braverman himself was critical of the debate about whether the average level of skills in society is rising or declining, as the answer leaves out the question of the polarization of skills by employing the criteria already in use to evaluate jobs as high or low skilled. There is a need to reconceptualize his thesis about skills and develop new criteria for the understanding of skills.

I seek to lift the debate out of the de-skilling vs. re-skilling confusion, developing an alternate set of analytical tools to make sense of skills, especially in view of the transformations associated with information technologies. I attempt to focus on deeper structures of skills independent of the question of whether de-skilling plagues all industries. For the moment, I also wish to bracket the question of who is "responsible" for de-skilling – managers, who seek to enhance their control, or profit-seeking investors whose investing decisions eventually force managers to make corresponding decisions. Design decisions about work and technology are socially shaped by a number of different and differing interests, including institutional inertia and resistance to change[28] and at times by mere chance. Technology is a human activity, which – at the level of actual practice – appears more like a "golem," a clumsy giant, than a superhuman agency with a clear road to unshakable ends.[29] Therefore, it is difficult to attribute changes in software designs, for instance, to the will of either scientists or managers or investors alone. The concept of skill saturation seeks to evaluate skills solely on the basis of their grammar and structures. Perhaps it would be expedient to clarify what this concept tries not to explain or analyze. It does not allude to the consciousness – obscure or obvious – of the manager or the worker, nor does it refer skills to the will of the work designers. The purpose of the thesis of skill saturation is not geared toward unearthing the hidden intent of work design, an intent that always goes beyond what is actually designed. It also avoids succumbing to the easy categories of low and high skills, or linking the skill level to one's income or hierarchical position. All of the above are valid research questions, but this investigation is specific to skills and their exhaustive or non-exhaustive grammar.[30] Despite this specific focus on skills, I do not explain skill saturation as a random phenomenon affecting isolated, unconnected pockets of work; rather, I understand skills in terms of the changing structures of rationalization.

Before discussing rationalization and skills, it is useful to throw light on the empirical material used and its purpose in this article. The basic argument of this study stands on theoretical grounds. I use quotations from my interviews with information workers solely for the purpose of illustration and clarity. These interviews are part of a larger research project on information technologies and the changing nature of work, being carried out by this author in California and New Jersey (U.S.A.) and Delhi (India) since 1995. Although the larger project comprises both survey and interview methods, the present analysis draws on 35 formal interviews – in addition to a number of informal conversations

– with a diverse group of workers, whose work involved some form of interaction with electronic text and graphics. These workers were selected from a variety of occupational groupings including secretaries, bank tellers, managers, civil, mechanical, and electrical engineers, hardware and software engineers, programmers, and system analysts. Contacts with informants were made mostly through snowball sampling; yet, only those workers were included in the sample that had directly experienced the introduction of new information systems in their workplace, and thus could distinguish what was *before* from *after* the introduction of new technologies.

Rationalization and skills

The rise of rationalization or instrumental reason[31] has been a distinguishing feature and a haunting theme of modernity.[32–34] My argument concerns itself with its two developments: first, the mathematization of experience, knowledge, and work, fashioning all life activity on the model of the natural sciences; and second, the growth of means-end rationality, which refers to an increasingly precise calculation of adequate means to attain a definitely given end, and the extension of such calculation to the "conduct of life" itself (Weber). In other words, rationalization implies constituting and ordering the world through the calculability of different phenomena, abstracting from the infinitely complex world a form of knowledge capable of its technical control and exploitation.

I extend the concept of rationalization to the constitution of skills. By rationalization of skills, I understand a two-pronged process of formalization and functionalization of skills; it is a process by which first, skills are analyzed into their parts, scientifically organized and standardized into well-defined steps, and then carefully codified. Second, while this process makes these skills an ever more precise and calculated means to a defined end, and turns them into more efficient instruments in the further control of environment and work, it also risks transforming skills into a ready-made program for the worker. Contrary to certain interpretations of rationalization, this study avoids, as stated earlier, the claim to any linear progression or decline of general skill level; nor does it imply that rationalized practices always win against less rationalized practices in the real world. Many scholars have pointed out how rationalization at times results in failures, glitches, problems, and other unintended consequences. Harley Shaiken,[35] for instance, describes

how a form of rationalization, viz., Total Operations Planning System in a certain firm, failed on the shop floor due to a combination of factors: technical flaws, dissatisfaction among production management, and shop floor resistance. Clearly, this study does not subscribe to the view that rationalization always succeeds in reality. Rather, it is more concerned with the imperatives of formal rationality that produce a whole series of effects in the real world by acting as grids for the perception and evaluation of things. The rule of formal law, for instance, may never be fully realized in actual sociopolitical situations, nonetheless it does act as the dominant principle and program of what Weber calls legal rational governance. Despite Weber's thesis of the Protestant Ethic as a moving force behind the rise of rationalization during the early days of modernity, it has proved rather difficult to specify exactly the factors behind the emergence and continuance of rationalization in modern times. What has been easier to specify is its widespread prevalence – in both socialist and capitalist regimes – as a way of organizing production and governance, as has long been discussed by critical theory. Practical achievements of the natural and applied sciences, resulting from dominant discourses of "productivity," "efficiency," and "formal logic" along with their associated practices may perhaps be cited as some of such factors. Broadly, rationalization is understood in this study as structuring the possible field of work skills.

An important contribution to the rationalization of skills was made possible by Frederick W. Taylor in the late nineteenth and early twentieth century. In his task management system, each worker was given a definite task with detailed written instructions and an exact time allowance for each element of the work. The task was based upon detailed time study, and standardization of methods, tools, and materials. All tasks were calculated, tabulated, catalogued, and reduced to rules, laws, and procedures. Skilled action was dissolved, in the industrial age, into a sequence of manual reactions to prescribed norms. Skills were subjected to external standardization, as they were motivated, guided, and measured by standards external to the worker. Individual differences were subjugated to an external routine, and the qualitative features of individual labor were quantified. Skills lost their intuitive, individual, internal character to an external calculable efficiency. Although there was a development of skills of a conceptual kind, like those possessed by engineers, many other skills were rationalized to triviality. As Veblen[36] put it: "The share of the operative workman in the machine industry is that of an attendant, an assistant, whose duty it is to keep pace with the machine process.... His work supplements the machine process rather

than makes use of it. On the contrary the machine process makes use of the workman." Taylorism emerged not only as a business ideology to legitimize managerial dictatorship in terms of efficiency;[17, 37] it also exemplified what Lewis Mumford[38] describes as an undue stress on techniques and technology in Western civilization – a line of thought later developed theoretically by Marcuse[32] and empirically by Noble,[7, 39] highlighting the historical roots of technological domination and mystification.

The rationalization of skills, in the context of information technologies, appears both a continuation as well as a radical shift from the earlier period. While it continues the tendency toward the efficient calculation and analysis – to the last detail – of each element of work, it also introduces a new element; that is, the conversion of various tasks into electronic symbols. The symbolic conversion of tasks in terms of text and graphics enhances the potential for re-arranging various elements of work in entirely new configurations. The ability to break down, digitize, and then repackage into a program the mathematical skills of an accountant, a civil engineer's mathematical and visual conception of a three dimensional structure, an architect's drawing skills, and some of the managerial decision-making skills – suggests a double orientation of new technologies. First, the tendency to insert logical structures into skills turns them into a more efficient means to a better manipulation of the world, thus reducing the possibility for play within the structure of skills. Second, by integrating previously separate(d) tasks and combining skills, the new digital medium of work has a potential – not fully realized yet – for increasing the space for play, a potential that is explored in the last section of the article. The integrated tasks in themselves do not offer freedom from skill saturation. By ordering and settling all ambiguity under a stringent program and reducing all irregular details of work, rationalization can give rise to saturation even among integrated tasks. A detailed exposition of the thesis of skill saturation will help clarify these points.

Skill saturation

The concept of "de-skilling" alludes, in one sense, to an important phenomenon – the loss of control over one's own creativity at work. Work, as Marx[40] pointed out, relates very significantly to what we are, and how we live:

> For labor, *life activity*, and *productive life* appear to man [*sic*] at first only as a means to satisfy a need, the need to maintain physical existence…. In the mode of life activity [however] lies the entire character of a species, its species-character; and free conscious activity is the species-character of man…. Man produces free of physical need and only genuinely so in freedom from such need.

The point here is not the essence of the human species discussed by Marx, but his stress on the non-purposive, non-instrumental character of work that transcends survival needs as well as imperatives of capitalistic enterprise. The skills used at work are not only tools to achieve work objectives, but also an expression of one's engagement with the world, a source of meaning and joy one feels just by executing those skills, a purposeless smile in the heart of purposive activity.

The creative aspects of skills trace their ontological roots to what Gadamer[41] calls the play mode of being, a mode in which all those purposive relations that determine a person's active, serious and caring existence are "curiously suspended"; play itself carries "its own, even sacred, seriousness." The freedom of playing oneself out is not realizable without transforming one's purposive behavior, or disburdening oneself of the stress of purposive mode, which constitutes the actual strain of existence. However, rationalization by its very nature derives its force from purposive rationality, which tends to functionalize every element of work and skills, and eliminate all purposeless spaces or gaps for play. To distinguish the skills that have completely been instrumentalized from the skills that still have some latitude for play, I introduce a distinction between "saturated" and "unsaturated" skills. Greater rationalization leads to what I call "saturated skills," in contrast to less rationalized, "unsaturated skills." The difference between the two is only of degree, not of kind.

To clarify the above terms, a surface analogy can be made with single-bonded *saturated compounds* in organic chemistry that are characterized by their inability to react with other chemical species, because all the four valences of carbon atoms are saturated by single bonds to four other atoms or groups. Their inertness and lack of chemical reactivity is due to their saturated nature; therefore, they are also known as *paraffin* (Latin, parum = little; affinis = affinity). The compounds in which carbon atoms are linked together by multiple bonds are known as *unsaturated compounds*. They are characterized by chemical reactivity due to the availability of extra bonds, and, thus, are able to form other compounds on addition of other chemical re-agents. In social theory,

any analogy with the terms used in the natural sciences is a hazardous task. I do not intend to enforce this analogy rigorously, or extend it beyond their metaphoric resemblance. The term "skill saturation" acquires its own independent meaning during the course of this discussion, especially after my comparision between ordinary and formal languages.

To continue the analogy, skill saturation refers to a process of rationalization, whereby skills move from the unsaturated to saturated state, from a multiple-bond to a single-bond structure of skills; that is, all the previously unanalyzed elements of skill and knowledge are analyzed, made explicit, and turned into a formalized procedure of singly bonded structure. They do not allow enough space (i.e., chemical reactivity) for play and creative uncertainty of outcome. Saturated skills imply an inertness, a plenitude, and a positivity that do not allow for a deep creative engagement with either the task or its local medium. An example of saturated skills is provided by a bank teller's detailed description of her job and required skills:

> You log on, do your password, then your screen opens … there're functions on the top, that say twenty-one is a cash advance, twenty-two is…, and it does nothing until you put in the number for the transaction you're going to do. Then there is a list of the amount – is it cash, is it check, does he want cash back … and [the relevant screen] pops up; if it's over a certain amount, another screen pops up and says, did you check ID. So, it's pretty basic, it takes you step by step through the transaction. It says, now give the customer this much money, and asks you if this amount is correct. And so you fill in numbers for all the sections, hit enter, it will take you to the next step. You will validate the thing you're holding – the check, the slip, the transaction – and then it asks if there is anything else you want to do for the customer. And you say yes, or no. Then it goes back to the first screen, which is just blank, and has the kind of transactions you can do. Sometimes, it's really slow. I was frustrated because I knew what was coming up, and I had to go ehhhhh, and had to wait for it to go through, and I knew that there could have been a faster way to do it, but I had to go through the steps, which was frustrating sometimes. It was so structured to take the person through it, to make it simple to prevent mistakes. [However] they are trying to make these programs faster.

The software used by the bank teller orders time in a single bonded, step by step structure that denies the worker any creative space of a conceptual or technical kind. Windows pop up in chronological time after each step has been completed, precluding any possibility for play, creativity, or experiment. Such software applications are not confined to banks alone; their use is quite widespread, ranging from banks, airports, state bureaucracies like DMVs, and hospitals to department

stores. Saturated skills, thus, are marked by a complete predictability of procedure and outcome. The intimate perception of skill, a tacit dimension of play, and the haziness of outcome are turned into a series of explicit procedures signaling in advance what is to follow. This leads to a job that is so highly defined and transparent that the skills used do not have a rift in their ordered plenitude, the rift for play and movement. With skill content made completely explicit, the creativity of the worker consists in following the fixed steps.

Here, we may re-visit Braverman's thesis of de-skilling in the context of saturated skills. Braverman's emphasis on the division of labor and fragmentation of tasks led scholars to question his thesis on the basis of task integration brought about by information technologies. Some scholars[42] argue that the banking tasks remaining to human tellers after the introduction of automatic teller machines combine many different skills, e.g., handling problems and inquiries, managing different types of accounts, and selling specialized services, that may require considerable knowledge of banking procedures, and skill at dealing with people. However, the thesis of saturated skill puts its emphasis on the lack of potential for play in a task rather than its fragmentation. In this context, the job and skills of bank tellers have not been de-saturated under information technologies, as discussed above; instead, the shift has occurred from one kind of saturated skills to another. First, the retraining of bank tellers for new skills – though a tiring affair for some of them – does not take more than a couple of weeks, and cannot be characterized as tremendous re-skilling. Second, the added dimension of the selling skills required of a teller does not add much to the potential for play either, due to its deeply purposive character, as voiced by another bank teller:

> I feel rejected, that's why, I hate selling, I can't stand it, when customers say flat out – no. I had someone say, don't try selling, pushing your sales tactics at me…. I have a hard time with that, and I also have a hard time with customers, who shout at me or scream at me, they think I am in control of what happens with their money, and I am not.

Although most software designed for the average worker in institutions as varied as libraries, hospitals, airports, insurance companies, and hotels has little room for play or imagination, the rationalization of skills does not explain all of work. Rationalization is only a general tendency, rather, a form of power that, despite informing all programs and projects of management, does not exhaust possibilities of resistance, or even failure. The resulting programs – always socially mediated – fail the utopia (or dystopia) of *total* rationalization. Sometimes organiza-

tions themselves find rigid computerized protocols dysfunctional, as they tend to hamper rather than facilitate task execution.[28] At other times, workers resist the built-in rigidity of software by various means. The following account of a bank teller highlights how workers resist rigid task performance and surveillance mechanisms by inserting an element of game into them:

> Teller: At the end of the day, you print out your whole journal, you log off, and it [computer] prints out all your transactions, not all your transactions, but … like if you wanted to cash a check, it will list how many cash paids I've done, and my average time for each cash paid…. So, [the computer] will list all the tellers that were logged on that day, and it'll list their transactions and their average time for all the transactions. Now, the officers when they close the bank, they also print out the stats, and that has everybody's average time.

Author: So, do you feel a kind of pressure on you?

> Teller: A little bit. I think they don't really do anything with the times, at least that's what they told me, but between the tellers, it was kind of a fun competition that we would always do. Like, what was your time today, and we would say, oh, all my times are under one minute…. And we started playing this game; we had certain rules like you couldn't be in the running – if you did less than one hundred transactions – to even count for that day, and then people started trying to go fast, and started making mistakes, and they sort of got mad at us, because they didn't want us to try and beat each other's time.[43]

Further, the real designs of software may not be as efficient as expected of calm and calculated programs of rationalization. Systems analysts corroborate the findings of Salzman and Rosenthal[28] that in reality institutions prefer to replicate and embed the design of previous systems into software despite the systems analysts' contention about their inefficiency. Yet, programs of rationalization – animated by discourses of productivity and efficiency – eventually produce real effects and transform practices. Corporate attempts to re-engineer organizations completely through systems such as Enterprise Resource Planning (ERP) exemplify efforts at avoiding the inefficiencies of earlier systems. Saturated skills always remain an actual and possible problem in projects of rationalization.

Unsaturated skills, on the other hand, tend to contain multiple bonds with the job and certain unanalyzed dimensions to allow enough room for action to take place, an action based on long and intuitive understanding. It implies engagement that is implicit, inherent, and defies clear visibility. Michael Polanyi's[44] concept of "tacit dimension" may allow us to understand how the unqualified process of the formalization of skills leads to complete predictability, and eliminates the element of

creative freedom and discovery. Polanyi explained tacit dimension as something that remains unanalyzable in an action. Zuboff (1988) quite lucidly captures this meaning of skill in her contrast of action-centered skills, which are less analyzable and deeply embedded in the sentient body, with intellective skills. However, the concept of skill saturation is conceptually independent from physical or intellective dimensions, and the distinction between saturated and unsaturated skills does not overlap with action-centered and intellective skills. In fact, some purely intellective skills can be shown to have an unsaturated character. For instance, in an intellective game of chess, a skillful player does not think in terms of calculation of possible, particular moves, but fuses multiple realms of instant pattern recognition, immense intuition, and a lifetime of experience. As a recent example, in a match between IBM's computer Deep Blue and Gary Kasporov, the skills of the machine were in total contrast with those of Kasporov. In contrast to a machine that evaluated 200 million positions per second, Kasporov – who could evaluate only 2–6 positions per second – competed on the basis of his unsaturated, multi-dimensional intellective skills. The player's perception, unlike the computer's calculation ability, is not completely analyzable and defies clear description. Meticulous detailing and computing of particulars – a logical path for the machine – may destroy our understanding of the subject and the joy we derive from it. Many creative skills are performed and learned by "indwelling" and "interiorization," rather than by explicit, formalized knowledge.[44]

Unsaturated skills have bonds and connections that house ambiguity and equivocation, resisting complete explication, or codification. They also remain tacit to a large degree. A surgeon's skills, for instance, depend not only on a clear knowledge of anatomy, but also on an intuitive understanding, immense latitude for improvisation, and unpredictability of outcome. Some of the managerial skills also have some tacit dimensions. Leadership qualities, ability to convince others about the soundness of a decision, an intuition about the consequences of a line of action – are all uncodified aspects of managerial skills. However, some of the decision-making skills are getting saturated with the ability of some software packages to measure the exact probability of different outcomes, where the manager is required merely to operate the software. Thus, the knowledge that underlies unsaturated skills is not fully formalized and functionalized; such skills tend to have elements of play embedded in them. Another example relating to culinary skills can illustrate the structure of both saturated and unsaturated skills. The skills of a cook who prepares a dish based entirely on a

ready-made recipe may be described as saturated, because the step-by-step procedure of the recipe book guides one through specified ingredients, their measured quantities, temperature control, and strict stages of cooking. One merely implements the written procedure in one's cooking. The cook who employs unsaturated skills, on the other hand, would rely more on tacit cues gained through experience such as the changing smell, color, and texture of the dish, an intuitive understanding of spice combinations, amount of ingredients, improvisation, and a vision of the final dish.

In information work, relatively less saturated skills consist chiefly of programming skills. Although programming languages are relatively saturated languages in terms of their highly procedural and sequential architecture, the task of programming, due to its problem solving nature, implies an implicit understanding of the problem and many possible ways of its solution, even if within the framework of logical rules. How exactly one is expected to arrive at the solution is not yet codified. According to a programmer: "Programming is really only problem solving. You need to identify the system of the problem, what are the parameters involved with it. You have to understand as best as possible the mechanisms that are involved with this particular system, and then you're ready to write your program."

On a spectrum from higher saturation to lesser saturation, this study identifies two kinds of programming. Programming for a definite task, and programming for an imagined outcome. The first implies a relatively saturated skill, as it consists chiefly of translating and codifying an already known task (e.g., an accounting procedure) into a software package, where all the elements of the system are known and the creativity consists in creating a simple and efficient program. Writing software for an imagined outcome, e.g., designing a chip with an imagined higher processing power, is, on the other hand, a task for which no procedure has yet been identified, implying a relatively unsaturated skill. As a computer scientist put it:

> The difference is only in the nature of the task. If you are doing some software for [which] there are tasks that have been done before, for the last ten years, and you know exactly the best way to go about it, and you could write software for that; let's call that kind of software "routine software"; and then you have other people, who are writing software to do things that haven't really been done before, and they don't know the best way of doing it. They hope it will work out well, and if it doesn't, then the next time over, when they write it, they'll improve it a little more. Those are usually called high technology tasks, and that's a pretty standard term that's used.

> There are high tech jobs and then there are just standard computer science jobs.

Whether for an "imagined" or a "definite" outcome, software writing suggests a feature of computer work that the thesis of skill saturation should necessarily take into account. I prefer to call it the "linguistic turn" of technology. Discussions of technology have perhaps been too focused on tools and tool contexts. Armed with computer languages, technological practices are experiencing a paradigm shift from a materialistic tool-making enterprise to symbolic and linguistic imagination, from technology-as-a-tool to technology-as-language. The linguistic basis of technologies is doubly significant. Software writing is not only the fastest and one of the largest growing occupations in the United States[45] with enormous influence on all work practices in general[46]; it is also a space where other technologies are imagined and invented. It is through software's remarkable capacity for simulation that new, more efficient electrical circuits, switches, bombs, and the more aerodynamic models of rockets, cars, and airplanes are tested before their actual production. The thesis of "skill saturation" will gain further crystallization when we closely examine what lies at the basis of computer work – programming languages.

The linguistic turn: A history of programming languages and skill saturation

In what respect could programming languages – such as BASIC, FORTRAN, Pascal, PROLOG, C++, or Java – be called languages? The history of computer software and – to a degree – hardware designs can be traced to a search for an "ideal language" – a language that is precise, unambiguous, and clear in structure, on the model of symbolic logic, in contrast to ordinary language, which is nebulous, misleading, and at times contradictory. The development of symbolic logic was intended to achieve a systematic closure of all equivocation, ambiguity, metaphorical play, and undecidability inherent in ordinary language. The goal thus was to reduce the operation of linguistic signifiers to single, unequivocal referents, and attain saturation in language with predictable accuracy. Gottfried Wilhelm Leibniz,[47] a seventeenth-century philosopher and mathematician, was the first to propose – perhaps after the Cartesian vision of universal mathematics – a "universally characteristic language" (*lingua characteristica universalis*). Such a language was supposed to represent concepts notationally

by showing the more fundamental concepts of which they were composed. Once converted into universally symbolic forms, such as graphs and diagrams, these concepts would be understood by all readers, irrespective of their native language. In the late nineteenth century, Giuseppe Peano[48] – with his work on Interlingua, an uninflected form of Latin – vigorously pursued the Leibnizian conception of universal language. The pursuit of a logical language also inspired Gottlob Frege, Bertrand Russell, and A. N. Whitehead, triggering the development of the logical language LOGLAN and the computer language PROLOG.

Leibniz also proposed a "calculus of reason" (*calculus ratiocinator*), involving exhaustive manipulations of symbols according to established rules. This would help either discover new truths or test proposed conclusions to see if they could indeed be deduced from the premises. Reasoning could thus mimic the operations of large sums in the manner of algorithms, and thus be free from individual errors and failures of creativity. Such reasoned derivations could be verified and performed by machines, which makes Leibniz the first to contemplate the possibility of a computer-like machine. Leibniz's idea for building machines to deduce valid references was taken up by Charles Babbage, William Stanley Jevons, and Charles Sanders Pierce and his student Alan Marquan in the nineteenth century, resulting in immense success of modern computers in the second half of this century. Although Leibniz's work predates George Boole's remarkably similar work by two centuries, the later is widely recognized for initiating symbolic logic. Boolean algebra – whether an independent work or not – helped create a binary system of logic, which turned out to be crucial for information theory. It constituted the basis for the circuit and transistor designs used in electronic digital computers.

In the twentieth century, the attempt to dispense with linguistic ambiguity is reflected in Wittgenstein's[49] early view about the role of language as providing a "picture of reality." Truth-value, under this view, consists in making logical propositions with one-to-one correspondence to reality. The early Wittgenstein and the early Russell imagined an ideal language that would act as a criterion for determining the meaning, or meaninglessness, of statements about the world. While the project of developing an ideal and logical language did not succeed in its original sense, especially when Wittgenstein[50] totally turned around in his analysis of language, the influence of symbolic logic retained its value. Russell and Whitehead,[51] in an attempt to prove that logic precedes mathematics, showed how all pure mathematics can be deduced from

certain ideas and maxims of formal logic, by the help of the logic of relations, without any undefined idea or unproved propositions.

Most of the rationally consistent approaches to information theory, following Shannon,[52] employ the structural rigors of formal logic, replacing the looser syntaxes, grammars, and vocabularies of ordinary languages with their symbolic, poetic, and surplus meanings. Cybernetic theory and computer technologies demand rigorous but uncomplicated languages to permit translation into non-ambiguous, special symbols that can be stored and utilized for mathematical manipulations. The closed system of formal language proved ideal for this need. Conclusions drawn using syllogisms according to logical rules could be tested in a consistent, scientific manner, as long as all parties communicating shared the rational premises employed by the particular system. The assumption of basic theorems of information theory is that the message transmitted is well organized, consistent, and characterized by relatively low and determinable degrees of entropy and redundancy.

All computer languages, informed by rigors of formal logic, may be seen as a result of rationalization – a certain formalization and functionalization of language, whereby the ambiguity and complexity of ordinary language are reduced to rationalized structures, with high predictability of procedure and outcome. Ordinary languages, on the other hand, are grounded in a metaphorical play of endless referencing without a possibility of formal closure. It does not mean that computer science is free of metaphors, for all language, including mathematics, is metaphoric – that is, an object of knowledge is always constituted in terms of its relation to something else, or else it would remain inexpressible.

The question is thus how differently are metaphors constituted in "ideal" and ordinary language, or in saturated and unsaturated linguistic structures. By my earlier definition of saturation, the question is relatively easy to answer. While formal language creates an artificial ground for metaphors, a place on which all fugitive meanings are strictly chained, ordinary language does not ground metaphors in a single referential domain, allowing free play and a certain undecidability to emerge. Students of symbolic logic in philosophy recognize this difference, when they encounter the difficulty of translating multidimensional ordinary language sentences into calculable formal sentences of symbolic logic, strictly tying metaphors to unambiguous meanings in single bonds. Although the ability to close interpretive and ambiguous spaces

for metaphoric play is an asset in software writing, it can also lead to certain problems. One basic problem with the traditional conception of artificial intelligence (AI) has been its understanding of the English language as possessing a certain necessary and fairly simple underlying structure, which could be captured through formally defined algorithms or data structures,[53] missing the later Wittgenstein's[50] important shift from his earlier position. The later Wittgenstein viewed language more like a device that can be used for indefinite ends. Any attempt to codify how it operates through a closed set of rules might lead us, for instance, to an absurd belief that there are necessary rules limiting the use of a vegetable knife to the act of cutting vegetables only, forgetting that the knife might as well be used for opening a cardboard box or loosening a screw. Ordinary language is a human institution that is governed – not by an external set of rules or a priori theories – but only by what people view as correct and incorrect in different contexts. In view of the above, Philip Agre[54] asks AI researchers to be reflexive about their underlying assumptions with an increased awareness of the role of metaphors in computer modeling. Despite these concerns, programming skills normally demand strict adherence to singly-bonded structures; for instance, using Structured Query Language, a programmer is bound to stick to a formally closed style of writing, such as "Select salary from payroll where employee = 'Jones.'" In short, the linguistic rationalization through symbolic logic is achieved by mechanically closing the play of surplus meaning and undecidability of propositions. It provides a close analogy and explanation for the concept of skill saturation, which is caused by a relative closure of interpretive space in a programmed field of work.

The structure of saturated skills offers a possible explanation for the paradoxical phenomenon identified by this study; that is, while information skills make it easy for the workers to perform their work, they also intensify their work experience. The high level of predictability enabled by skill saturation affords greater control and ease for task execution, which is further enhanced by the reduction in spatial and physical commitments. Yet, as we discuss later, the greater ease and control does not imply the lack of intensity in work performance.

Work made easy

The new skills, with reduced physical demands on the body, seem to empower the worker. Industrialization was believed to have dethroned

the worker from the center and established the rule of the machine. In the information age, the situation appears to have changed with the consolidation and integration of tasks through information technologies,[20, 55] where the worker can perform a variety of tasks sitting at a desk with unprecedented ease. At a clerical level, customer service representatives can attend to all aspects of a client's transactions: taking orders, entering data, making adjustments, and answering inquiries.[42] The toil, the repetitive routine labor, the time-consuming physical movements seem to have been done away with. This understanding is shared by all informants, ranging from secretaries to computer professionals, who think they are able to perform their work with more ease:

> It's just great having all that information in there, because people are always asking me about statistics, different kinds of information … and Quattro Pro is very useful. I can't imagine trying to keep track of all that money on paper. And it's kind of neat, all these databases are all linked; all these magical things keep each other updated. So, it's really nice to keep everything at hand now.

It has become possible for the worker to dive deeply into the content of work, and perform the job with relative ease. The development of simulation technologies has opened up new fields of possibility for the engineer. A hardware engineer can develop and test a new chip without having to deal with the real hard medium of wires and transistors, just as an electrical engineer can control and design electrical equipment without having to deal with heavy electrical machines and transistors. As a computer scientist puts it, with the rise of information technologies, the nature of hardware engineering has dramatically changed:

> [The job of a hardware engineer now] is mostly a software job. They say you're a hardware engineer, but you're actually out there writing programs to build the chip. In the olden days, it used to be a proper hardware job, which means that you take a soldering iron and wires, and you would fix them yourself. Now you sit in the front of your computer and you have these little pieces of software, which help you lay down wires on your screen; you don't have a real wire in hand, and you put transistors here and wires there. Then you simulate it to see if it works.

Simulation techniques enhance the power of the engineer manifold. Unlike material objects that always appeared in a spatial mode constituting the earlier world of a hardware engineer, information disclosed through software does not occupy space in the same sense. All dimensions of work appear on a screen that acts as a window to the job being carried out in virtual space. Work becomes the simulation of work. And space can be infinitely produced on the screen as simulated

spaces. Elaborating upon the concept of simulation, my informant told me:

> Suppose … I want to build a car. There are two ways you could actually do it. You could actually build a car and see if it works. Or you could build a fake car … in the mind of your computer, and then ask the computer to see if it works. The second way is better because you don't waste money, buying material. If you are doing a space shuttle, for instance, they can't afford to send five space shuttles to figure out some mistakes first. So, they simulate everything inside a computer to see if it will work.

Further, the integrative capacity of information technologies facilitates many different kinds of work. In the words of a computer professional:

> With the advent of the Internet, direct connections are possible. Like right now, I have a guy working for me in Prague. And it's very effective, because he can move fairly substantial quantities of data over the Internet; e-mail is essentially instantaneous. I can e-mail him; he'll receive it in five minutes.

While this development may potentially preclude the need for direct interaction at work, posing the problem of indirect and systemic integration of work relationships,[56] it offers the worker an extended reach over work domains. Information systems, which encompass a whole range of technological innovations, introduce with unprecedented transparency the job to be performed and the ease with which it can be completed. Software technologies allow higher end-use interaction and easy accessibility reflected in the evolution from the early esoteric machine code to today's high level, object oriented languages and PC based productivity tools. Software has become easier to conceptualize, build, and use.[57] As an administrative secretary puts it:

> When we got Wang [word processor], we realized the power, even though it was primitive compared to what we have today; we would end up having five, six, seven, eight revisions of a paper, whereas before we would get almost a final copy. And that was a huge difference.

The discussion of the worker's ability to produce enormous quantities of data and control various aspects of work with relative ease leads us to discover the paradox of the simultaneous intensification of work.

Work made intense

In many cases, the newly achieved power rebounds on workers by an intensification of work, which appears as one of the effects of skill saturation. By reducing space for free play, saturated skills offer an

extremely tight frame of reference within which one is supposed to function and execute. In the industrial context of scientific management, for instance, time-motion-based methods produced skill saturation by restricting the bodily movement to purposive efficiency-oriented action. The body's interpretive space for physical action was closed through externally limited choices. Skill saturation in information work leads to intellective intensification, where a programmed and programming logic creates a demanding and unyielding field of grammar for all intellective action to take place. The earlier comparison between ordinary language and formal language disclosed how the closure of equivocation and interpretive space inherent in an ordinary language is the very goal of formal language. Yet, the relative loss of interpretive freedom and strict demands of symbolic language without a license to break rules – all tend to stress the need for unflinching intense intellective attention. The rules do get broken and mistakes are bound to occur, but the very demand for accuracy required by new skills is complicit with work intensification.

The intensification of information labor consists in morbid attention to detail that most skilled jobs, ranging from a bank teller's to a software engineer's, require. With reduced physical obligations, the new skills demand one's unwavering intellective devotion. As a bank teller put it: "You have to watch your inputs, the cash you take in and the cash you take out; you have to watch that like a hawk, and it's very easy to make mistakes; even for the people who have been there forever." Constant attention is demanded of nearly all information work. A software engineer is no exception. As Ullman,[58] who worked in software development for fifteen years, writes:

> When you program, your mind is full of details, millions of bits of knowledge. This knowledge is in human … [rather chaotic] … form. To program is to translate between the chaos of human life and the line-by-line world of computer language…. You must not lose your own attention. As the human-world knowledge tumbles about in your mind, you must keep typing, typing. You must not be interrupted. Any break in your listening causes you to lose a line here or there … you will create a bug and there's nothing you can do about it.

The new skills are invested with a new logico-mathematics of power, characterized by abstract mathematical logic, requiring one to think in a specific, disciplined, procedural, and sequential way. As one of my informants put it:

> Programming has been procedurally oriented…. It's very sequentially oriented. With the advent of the object oriented programming, people are making a claim that it's diverging greatly from the procedurally oriented languages of

> the past. In a way that's true, but in a way it's not true, because still when you are solving a problem, things still happen in a sequence.

Another aspect of work intensification relates to the intensification of time consciousness. Although the time taken in the process of job execution is close to instantaneity with the delay of only a few seconds, the chief concern of employees is that computers are not fast enough. Computers are always found to be lagging behind the mind's capacity to move forward. It requires the constant replacement of old systems and old software with faster versions, which also occasions what some have called re-skilling in newer versions. It implicates humans in such a way that they themselves want information to move faster, thereby increasing the possibility of exploitation by time-centered technologies. Here is one bank teller describing her experience of technologies as being too slow: "It's ridiculous … it was only recently that we were able to get even 486s in our PCs, so that we could do our inputs and outputs faster; we used to have to key in a transaction and wait a couple of seconds. It shouldn't be that way."

Space being less relevant, it is time that becomes all important. A short lag between commands given and commands executed occasions frustration, as there is nothing else to engage in during the agonizing moments of wait. The loss of immediate connection of skill execution with its spatio-material context[59] and the introduction of phenomenal speed further intensify work experience. During interviews and conversations, most of the workers felt that their workload and workplace in effect has rather increased over the years due to computers. In reply to a question why she thinks people have less time now in the office, one secretary said: "Because they are busy. Let's put it this way. I couldn't do this job, if I didn't have a computer. Yes, it's a time saver, but that just means you take on more work."

The slow movement of earlier technologies did not encourage certain kinds of work and activity, which have been made possible through new technologies. Industrial skills still required physical movement as part of people's job. Information technologies, due to their very nature, reduce the need for movement in space, while making it possible for the work to reach the worker's desk instantaneously. They are the technologies of extraordinary speed and unprecedented immobility. As in the Fordist assembly line production, where it was the object of work that flowed from one place to another, while the worker was stationary, information technologies freeze the individual in the place

of work, while the work moves at the speed of light. This rationalization achieves a two-fold objective: economy of movement and extended organizational reach. First, it saves time and money spent on physical movement and traveling; second, the organization now can reach places, where earlier it was not possible to monitor certain activities, or send enormous quantities of data. A computer professional describes how one did the related work before the advent of the Internet and e-mail: "Oh, you didn't. I mean if you needed to have a meeting with someone, you flew out to them, and essentially we didn't have diverse organizations, because it wouldn't have worked very well."

A good example of both the reduced movement and extended reach is the rapid growth of telecommuting both within and between nations, reducing the need for physical migration and movement. Many informants described how software professionals, working for Indian-based companies overseas, are increasingly involved in developing software systems for U.S.-based corporations off-shore, upgrading their systems, delivering software packages, and providing maintenance by accessing their mainframe computers online across the globe.

The reduction of physical movement in the context of skills is more drastic in one sense and less so in another sense. While new technologies help reduce the need for movement in the actual performance of computer skills, they also have some mobility requirements akin to earlier technologies, such as travel requirements of field service engineers for the repair and maintenance of software and hardware systems at the equipment site. Yet, over the years, due to increased reliability of computer systems, field service organizations are witnessing a decline in annual service contract revenue, despite the continued growth in the installed base through new sales.[28] This reduced mobility of workers coupled with the high mobility of information results not only in the dispensability of a part of the earlier workforce, but also leads to the intensification of work for remaining workers, as a result of what Harvey[60] calls "space-time compression," in a new round of capitalist expansion and the new régime of "flexible accumulation." With distances turning transparent, a new form of space compression reduces the need for movement from one place to the other, and time-compression makes more work available at any one point of time. In addition to intensification, skills are undergoing another transformation – a growing conversion of different skills into computer skills as well as the relative lack of differentiation among new skills. Industrial processes proliferated a variety of skills because of their heavy emphasis on

specialization. Information skills, on the contrary, seem to move toward increasing correspondence through a more sophisticated rationalization.

Homogenization of work medium and convergence of skills

Earlier I discussed how computer languages owe their existence to a search for an ideal universal language with logical accuracy and algorithmic computability. In an important sense, computer-based technologies mark a departure from previous technologies – a departure, as noted earlier, from tool-making contexts to linguistic contexts. The new technologies enable an imagination that does not immediately need or produce empirical tools. The practice of simulating and designing an imagined car, or a space-shuttle, on a computer, and virtually testing its aerodynamic viability before actually producing it, turns the face of technology from technical tinkering to hypothetical modeling and symbolic manipulation. It is not surprising that the development of the digital computer itself relates to a hypothetical computing device – called the Turing Machine, which was not a real machine, but an idealized mathematical model, devised by logician Alan M. Turing. The model reduced the logical structure of any computing device to its core components by extrapolating the essential features of information processing. The Machine provided the basis for all subsequent digital computers, which share Turing's basic design of an input/output device (tape and reader), memory, and central processing unit (control mechanism).

Animated by a search for a universal linguistic model, the development of symbolic logic and programming languages liquidates some of the earlier particularities of technologies. Information technologies by nature recode in binary digits all possible types of tasks that earlier existed in different forms. Due to its capacity for universal coding, software can be developed not only for white collar jobs, but also for controlling heavy machines in industrial plants, thereby occasioning a homogenization of the medium of work across different jobs and occupations. Unlike earlier technologies that were confined to specific occupations and industries, the development of information technologies is affecting and transforming a vast number of occupations and unrelated industries. The new technologies are proving to be the language of all work. The new pervasive medium of electronic texts and graphics does not display an ether-like neutrality – like any other

medium – toward what gets performed through it; rather, in Marshall McLuhan's[61] "medium is the message" sense, it constitutes work in specific ways. Consequently, many previous methods of doing work have become redundant, and are in the process of being abolished and replaced by less differentiated forms of work performance. The following description of the heterogeneity and division of work prevailing in the 1950s described by C. Wright Mills is in need of a revision:

> In the enormous file of the office, in all the calculating rooms, accountants and purchasing agents replace the man who did his own figuring. And in the lower reaches of the white-collar world, office operatives grind along, loading and emptying the filing system; there are private secretaries and typists, entry clerks, billing clerks, corresponding clerks – a thousand kinds of clerks; the operators of light machinery, comptometers, Dictaphones, addressographs; and the receptionist to let you in or keep you out.[62]

With the introduction of advanced information systems at the workplace, "a thousand kinds of clerks" are giving way to de-layered and flatter organizational structures – a phenomenon viewed both positively in terms of lean, decentralized, "post-bureaucratic" management[20, 63, 64] and negatively in terms of displacement of workers and temporization of work.[65–67] While the logic of rationalization during the industrial period demanded specialization and task splitting for efficient, calculated action, its imperatives in the information age have changed to task merger and integration to attain similar objectives.

The homogeneity of the work medium is inextricably connected with the growing convergence of work skills, where the term "convergence" carries two compatible meanings. First, previously differentiated skills converge in a single, integrated software. For instance, software systems integrate many clerical skills – such as typing, editing, and formatting – into applications used by higher level professionals, thereby making specialized, separate secretarial positions less useful. Second, new skills are converging in terms of the similarity of the logic used to perform work. As more and more areas of work are being brought into the realm of computers, the new skills required of the previously heterogeneous workforce comprise the ability to think in terms of computer logic and master pre-given logical structures common to different kinds of software. Consequently, there is enormous boundary crossing in post-industrial work settings. For instance, an informant noted, a large number of Physics Ph.D. students are being hired by Wall Street-based firms not for their knowledge of Physics; instead, they are recruited on the basis of their keen logical ability, which makes it relatively easy for

these students to learn Structured Query Language and develop financial database software. To employ another illustration for this aspect of skill convergence, individuals working in areas as varied as hardware and software, electrical and mechanical engineering, process engineering, and simple programming, whose earlier jobs and training greatly differed from each other, are now writing software for different purposes. In addition to the earlier remark of the hardware engineer, whose job now involves only software engineering, here is an electrical engineer describing his job:

> Even ten years ago, most electrical engineers had very little to do with software. In fact, most electrical equipment had no software connected to it, but over the years, a lot of things have shifted to software … most things have become digital, and once things become digital, they can be controlled using a computer. So, slowly, everything is shifting to computers. A lot of electrical design is now being done on computers, because it can be done much faster. Yes, over the last ten to fifteen years, there has been a very big shift in electrical engineering. Now, every electrical engineer must know programming languages, which wasn't the case some years ago…. If I were to start [my career] again, I'd start more so in computer science, because right now, I've slowly drifted from hard-core electrical engineering to software. So, if I were to start again, I might as well start in software, than waste my time doing electrical machines and controls.

Some engineers reported that there is not as much need now for theoretical competence as previously required, because basic skills tested on the job relate to programming and logical skills. This tendency to *convergence* – as opposed to the *differentiation* of skills engendered by the industrial period – reflects a different form of rationalization. With the homogenization of previously differentiated domains of work, information skills divide themselves – at a general level – into two broad groups of workers: software users and software writers. All my informants belonging to diverse occupations were either using software programs or writing them. This distinction does not enjoy a strict boundary, as a software user writing macros is also a software writer at a reduced level of complexity. Yet, the distinction may be useful for a clearer grasp of one's task orientation and specific skill requirements.

The "convergence" of skills certainly does not imply an "identity" of skills. A secretary using a database program is not executing the same set of skills as someone using a statistical software package, for the latter requires a certain understanding of statistics principles. Yet, a statistics software application relieves the worker not only from the burden of countless calculations, but also – to a certain degree – from

the knowledge of mathematics that goes behind such calculations, bringing the two cases to a more comparable plane. Despite distinctions among various information workers, the overall convergence of skills across different occupations results from constituting work as a problem of coding, and of further coding within codes, a sort of symbolic sedimentation. This idea is well expressed in the phrase – "closer to the machine" – used by software engineers to distinguish those who deal with the foundational or deeper layers of machine code from those who use higher-level programs called "applications." The structuring of work as a general problem of coding not only creates conditions for the homogenization of the work medium and convergence of skills; it also enables thorough standardization within and across coded packages (software) to address the need for their compatibility and familiarity with the user. In view of this constant coding, standardization, and continued discourse of rationalization, we must explore if there are possibilities of resistance to saturation in post-industrial work.

Skill convergence and possibilities of de-saturation

The phenomenon of skill convergence seems to hide possibilities of both higher and lower saturation. On the one hand, there is a threat of saturation if various skills are converted into coded templates of work, where one's skills consist in merely knowing the menu commands or screen icons to perform work. On the other hand, there are also some promising cases, such as graphic designing, where mere knowledge of computer commands is not enough, and where there are not many ready-made templates available for work. One must employ multi-dimensional skills to create an aesthetic object on the screen with an intuitive understanding of colors, shapes, and composition without a template. In this case, what has been re-coded in the software program is not the artists' skill set, but only their tool set, which makes direct engagement with the world of virtual objects possible. Pelle Ehn[68] and Malcolm McCullough[69] both attempt to advance our understanding of computer-mediated work in terms of designing computer/digital artifacts. While Ehn proposes the language of artifacts in order to develop novel approaches to computer work for emancipatory purposes, McCullough maintains that recent advances in technologies have actually regained some of the holism of earlier craftwork that was lost during the industrial age. In the craft, manual and conceptual skills – a certain coordination of hands, eyes, and the mind – were harmoniously combined in direct manipulation of real objects. McCullough

claims software programs such as MacPaint and MacDraw, in the 1980s, were the first direct manipulation programs, requiring hand-eye coordination instead of typing numbers on a keyboard. Through the rise of such tools as Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) systems along with a number of other software such as graphic and paint programs, McCullough argues, a new technology is born "with old roots." Although it is true that we cannot boast of the "direct manipulation" of material objects as we did in traditional craft, we have regained some of its aspects in what may be called "indirect manipulation." Instead of being directly manipulable, our tools – symbolically represented in different software – can be conceptually and indirectly controlled. In a paint program, for instance, one can look around for a paint or brush, pick it up, hold it, and move it into relation with other objects. Such software, McCullough suggests, should be understood as a *mechanism* in our endeavor to produce what he calls "digital artifacts." CAD and CAM as mechanisms, according to him, seem to converge with traditional artisanry in terms of creating three-dimensional things in a tightened loop between design and fabrication. Just as in traditional craft, a good design was grounded in fabrication and vice versa; in CAD/CAM system, both design variations and fabrication process criteria drive one another. In this coupling, "input to physical fabrication operation is symbolic, and the output from geometric derivations is tangible.... Thus, after two centuries of separation, the conception and the execution of everyday objects are once again in the same hands."[69] Some scholars do not completely subscribe to the above view. In many CAD applications, due to the problem of countless new design alternatives, the designer is increasingly required to select merely from a "menu" of "optimized" alternatives, which may throttle rather than expand creativity.[35]

The idea of treating computer-mediated labor as mostly design-oriented work also suffers from the problem of exaggerating what is still a tiny part of IT work. CAD and CAM are used by a small fraction of regular information workers to make their use a general case. In the context of the United States, where the manufacturing sector is not the largest employer, most workers have not had any interaction with CAD and CAM. The majority of IT labor still consists of some sort of data entry and data manipulation work, including the work performed by bank tellers, accountants, secretaries, and all others who serve at the front and back ends of state and corporate bureaucracies. Generally, the software designed for the average worker in a variety of institutions has little room for play, imagination, and holistic harmony of craft-

work. A computer professional-turned manager, on being asked what strategies he adopts to make work more playful and interesting for his workers, replied as follows: "I never considered designing work to make it more fun. The idea didn't occur to me. My objective has always been to create better projects and complete them on time."

Despite the above discourse of productivity and efficiency prevalent in workplaces, we may conceive of three possibilities in skill, non-skill and socio-political contexts, where lower saturation may be attained.

Skill Context: Within the strict context of skills, there are two (or possibly more) situations where lower saturation is either a reality or a distinct possibility. In the first situation, logical structures of programming languages transform only the users' tools, and not their skills. One of the examples, as mentioned earlier, would be graphic designing skills that offer play spaces in such tasks as digital manipulation and enhancements of photographs, movies, print, and web pages. Despite the increasing availability of ready-made templates in many software applications for web page designing and newsletter publishing, the majority of such work is free from strict predictability of outcome, i.e., final product. Other related examples are what McCullough discussed under indirect manipulation through CAD and CAM applications. Similar applications are allowing architects not only to create better products (as quality products can be made in saturated skill contexts as well) but also to play with more possibilities. The second situation relates to certain programming tasks themselves. As I suggested earlier, software writing may consist of either programming for definite tasks or programming for imagined outcomes. The second case seems to entail less saturation as compared to the first one. A software engineer designing a faster chip enjoys considerable spaces for play, as the final shape of the chip is at best imagined, unlike the programmer who translates an already existing account procedure into software. In addition to the immediate context of skills, de-saturation can also occur in the larger sphere of work.

Non-skill Context: For a nuanced understanding of skills, we may discuss certain non-skill factors that reduce the intensity of saturated skills. While many workers may not be able to resist structures of skills themselves, they overcome the problem of skills through other means. For instance, in one of my examples, bank tellers sought to overcome the saturated skill structure by injecting into their work an element of competitive sports ("who can do it faster?") that was not required of

them. Although the structure of skills remained saturated due to its strict procedure, its intensity was suddenly reduced. Occasions of such resistance may easily be found in the whole range of office behavior: e-mailing, instant messaging, and playing and devising new computer games during work time. The hacker culture in the world of computing coupled with movements for free, open-source software (e.g., the Linux movement) have long been the forces of resistance not only to corporate imperatives, but also to the taming of their work culture. The resistance to saturation may also take place by direct intervention through sociopolitical means.

Socio-Political Context: Many scholars have emphasized the importance of sociopolitical resistance for the shaping of new technologies.[15–17, 39, 70] Alternative production movements, emphasizing the need for human-centered technology designs, are sites for resistance to skill saturation. Pelle Ehn's[68] discussion of technology designs based on qualitative judgments of workers and designers and significant trade union participation; Harley Shaiken's[35] argument for "A Technology Bill of Rights"; and Philip Agre's[54] discussion of a broader movement toward post-Cartesian conceptions of computational activity in Artificial Intelligence research – are a few such examples of social and discursive interventions against skill saturation.

Conclusion

The concept of skill saturation provides a way to re-conceptualize actual and possible problems concerning work skills in industrial and post-industrial contexts as well as a means to evaluate them. It offers a theoretical axis to understand work not merely in terms of higher or lower level of skills, blue or white-collar skills, but also of the relative absence of spaces for play, creativity, or engagement in the structure of skills. Skill saturation may occur due to a complex of rationalizing forces, acting as limiting factors on the execution and experience of skills. It results chiefly from the exhaustive ordering of task execution in a formal and explicit form. Even when one can choose from many possible steps to carry out a task, all steps in a path may already be pre-given or programmed in advance. The allowance or dis-allowance of equivocation and ambivalence is directly connected with the production or reduction of play in skills. Further, the likelihood of saturation increases with the intensive use of only one dimension of the material-intellective complex. The total reduction of skills to either bodily dimen-

sion (factory work) or mental dimension (information work) may potentially reduce spaces for play and pleasure of work. In addition, saturated skills, as a result of pervasive instrumentality, are marked by a strict predictability of procedure and outcome, offering both the comfort of certainty and inescapable intensity of programmed logic. Such predictability is enhanced by standardized symbolic coding of tasks as ready-made packages of rational software templates.

The enormous flexibility achieved through the symbolic conversion of a variety of tasks into software programs requires a smaller set of standard skills to be used for job performance, leading to the convergence of skills used across different kinds of work and occupations. The resulting skills can potentially be both saturated or unsaturated, but if not resisted, they have a potential to move on a very specific and one-dimensional path, a path decided by the logico-mathematical power of relentless rationalization.

Acknowledgments

An earlier version of this article was presented at the Annual Meeting of the American Sociological Association in Toronto, 1997. I am indebted for the helpful comments received from Erica Bornstein, József Böröcz, Karen Cerulo, Lee Clarke, Ira Cohen, James Ferguson, Eric Kaldor, David Smith, Judith Stepan-Norris, Douglas White, and the Editors of *Theory and Society*. The Social Science Research Council and the Population Council supported the larger research project that informs this study.

Notes

1. The Economist, *The Hitchhiker's Guide to Cybernomics*, in *The Economist: A Survey of the World Economy* (1996): 3–46.
2. Manuel Castells, *The Rise of the Network Society* (Cambridge, Mass.: Blackwell, 1996), 276.
3. Harry Braverman, *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century* (New York: Monthly Review Press, 1975).
4. Dan Clawson, *Bureaucracy and the Labor Process: The Transformation of U.S. Industry, 1860–1920* (New York: Monthly Review Press, 1980).
5. Joan M. Greenbaum, "In the Name of Efficiency: Management Theory & Shopfloor Practices in Data-Processing," in David Knights, David Collinson, and Hugh Willmott, editors, *Job Redesign* (Aldershot: Gower, 1985).

6. Philip Kraft, *Programmers and Managers: The Routinization of Computer Programming in the United States* (New York: Springer-Verlag, 1977).
7. David F. Noble, *Forces of Production: A Social History of Industrial Automation* (New York: Knopf, 1984).
8. M. Wallace and A. L. Kalleberg, "Industrial Transformation and the Decline of Craft: The Decomposition of Skill in the Printing Industry, 1931–1978," *American Sociological Review* 47/3 (1982): 307–324.
9. Kenneth I. Spenner, "Deciphering Prometheus: Temporal Change in the Skill Level of Work," *American Sociological Review* 48/6 (1983): 824–837.
10. Roger Penn, *Skilled Workers in the Class Structure* (New York: Cambridge University Press, 1984).
11. Carol Axtel Ray, "Skill Reconsidered: The Deskilling and Reskilling of Managers," *Work and Occupations* 16/1 (1989): 65–79.
12. Stephen Wood, *The Degradation of Work? Skill, Deskilling, and the Labour Process* (London: Hutchinson, 1982).
13. Paul Attewell, "The De-Skilling Controversy," *Work and Occupations* 14/3 (1987): 323–346.
14. Paul Attewell and James Rule, "Computing and Organizations: What We Know and What We Don't Know," *Communications of the ACM* 27/12 (1984).
15. Richard Edwards, *Contested Terrain: The Transformation of the Workplace in the Twentieth Century* (New York: Basic Books, 1979).
16. Andrew L. Friedman, *Industry and Labour: Class Struggle at Work and Monopoly Capitalism* (London: Macmillan, 1977).
17. Michael Burawoy, *Manufacturing Consent: Changes in the Labor Process under Monopoly Capitalism* (Chicago: University of Chicago Press, 1979).
18. Michael D. Yates, "Braverman and the Class Struggle," *Monthly Review* 50/8 (1999): 2–8.
19. Nicholas Negroponte, *Being Digital* (New York: Alfred A. Knopf, 1995).
20. Larry Hirschhorn, *Beyond Mechanization: Work and Technology in a Postindustrial Age* (Cambridge: Mass.: The MIT Press, 1984).
21. Shoshana Zuboff, *In the Age of Smart Machine: The Future of Work and Power* (New York: Basic Books Inc., Publishers, 1988).
22. Paul S. Adler, *Technology and the Future of Work* (New York: Oxford University Press, 1992).
23. A. Allan Hunt and Timothy L. Hunt, *Human Resource Implications of Robotics* (Kalamazoo, Mich: W.E. Upjohn Institute for Employment Research, 1983).
24. Cynthia Cockburn, *Brothers: Male Dominance and Technological Change* (London: Pluto Press, 1983).
25. Bruce Nissen and Peter Seybold, "'Labor and Monopoly Capital' in the Labor Education Context," *Monthly Review* 46/6 (1994): 36–45.
26. Peter Meiksins, "'Labor and Monopoly Capital' for the 1990s: A Review and Critique of the Labor Process Debate," *Monthly Review* 46/6 (1994): 45–60.
27. Joan M. Greenbaum, "On Twenty-Five Years with Braverman's 'Labor and Monopoly Capital,'" *Monthly Review* 50/8 (1999): 28–33.
28. Harold Salzman and Stephen R. Rosenthal, *Software by Design: Shaping Technology and the Workplace* (New York: Oxford University Press, 1994).
29. H. M. Collins and T. J. Pinch, *The Golem: What Everyone Should Know About Science* (New York: Cambridge University Press, 1993).
30. Another important issue that could not be discussed in this article is gender and its relation to skill transformations, as demonstrated by a vast literature on gender and

technology. I did not think it possible – within the framework of this study – to do justice to the gender component of information technologies, which requires another set of arguments and literature review.

31. This statement relates to the central concern of the critical theorists of the Frankfurt School, which may be expressed as the critique of the domination of instrumental reason in modern life, as contained in the School's earlier works Dialectic of Enlightenment. The School's early thinkers, viz., Max Horkheimer, Theodor Adorno, and Herbert Marcuse, for instance, agreed with Max Weber that the emergence of instrumental reason must be traced to pre-capitalistic ideas and modes of life, that the advance of instrumental reason led to disenchantment (particularly after Enlightenment) and to the undermining of traditional world views. But they adopted a Marxist approach to show how capitalism provided an impetus to the further development of instrumental reason.
32. Herbert Marcuse, "Industrialization and Capitalism in the Work of Max Weber," *Negations: Essays in Critical Theory* (Boston: Beacon Press, 1968), 201–226.
33. Stephen Kalberg, "The Rationalization of Action in Max Weber's Sociology of Religion," *Sociological Theory* 8/1 (1980): 58–84.
34. Stephen Kalberg, "Max Weber's Types of Rationality: Cornerstones for the Analysis of Rationalization Process in History," *American Journal of Sociology* 85/5 (1980): 1145–1179.
35. Harley Shaiken, *Work Transformed: Automation and Labor in the Computer Age* (New York: Holt Rinehard and Winston, 1985).
36. Thorstein Veblen, *The Instinct of Workmanship and the State of the Industrial Arts* (New Brunswick: Transaction Publishers, 1990 [1914]), 306–307.
37. Craig J. Calhoun, *The Question of Class Struggle: Social Foundations of Popular Radicalism During the Industrial Revolution* (Chicago: University of Chicago Press, 1982).
38. Lewis Mumford, *Technics and Civilization* (New York: Harcourt Brace & World, 1963 [1934]).
39. David F. Noble, *America by Design: Science, Technology, and the Rise of Corporate Capitalism* (New York: Knopf, 1977).
40. Karl Marx, *Economic and Philosophic Manuscript of 1844: Writings of the Young Marx on Philosophy and Society*, edited by Loyd David Easton and Kurt H. Guddat (Garden City, N.Y.: Doubleday, 1967), 294.
41. Hans Georg Gadamer, *Truth and Method* (New York: Seabury Press, 1975), 102.
42. Roslyn L. Feldberg and Evelyn Nakano Glenn, "Technology and the Transformation of Clerical Work," in Roslyn L. Feldberg and Evelyn Nakano Glenn, editors, *Technology and the Transformation of White-Collar Work* (Hillside, NJ: Lawrence Erlbaum Associates, Publishers, 1987), 77–98.
43. A comparison with Michael Burawoy's (1979) critique of the labor process as a shop floor game can be made here. Explaining his idea of manufacturing consent, Burawoy notes that the workers are first compelled to play the game; then they ironically proceed to defend its rules. In the case above, however, the game has a narrower context, as it was not developed or encouraged by the establishment.
44. Michael Polanyi, *The Tacit Dimension* (London: Routledge & K. Paul, 1967), 18.
45. *Occupational Outlook Handbook, 2000–2001* (United States Bureau of Labor Statistics, 2000).
46. Sheila McConnell, "The Role of Computers in Reshaping the Workplace," *Monthly Labor Review* August (1996): 3–5.
47. Gottfried Wilhelm Leibniz, *Logical Papers* (Oxford: Clarendon, 1966).

48. Giuseppe Peano, *Interlingua* (Torino: Academia pro Interlingua, 1927).
49. Ludwig Wittgenstein, *Tractatus Logico-Philosophicus* (New York: Harcourt, Brace & Company, 1922).
50. Ludwig Wittgenstein, *Philosophical Investigations* (New York: Macmillan, 1972).
51. Bertrand Russell and Alfred North Whitehead, *Principia Mathematica* (Cambridge: The University Press, 1925).
52. Claude E. Shannon, Warren Weaver, *The Mathematical Theory of Communication* (Urbana: University of Illinois Press, 1964).
53. Hubert L. Dreyfus, *What Computers Still Can't Do: A Critique of Artificial Reason* (Cambridge, Mass.: MIT Press, 1992).
54. Philip Agre, *Computation and Human Experience* (Cambridge: New York: Cambridge University Press, 1997).
55. Paul A. Strassmann, *Information Payoff: The Transformation of Work in the Electronic Age* (New York: Free Press, 1985).
56. Craig Calhoun, "Indirect Relationships and Imagined Communities: Large-Scale Social Integration and the Transformation of Everyday Life," in Pierre Bourdieu and J. S. Coleman, editors, *Social Theory for a Changing Society* (New York: Russell Sage Foundation, 1991), 95–121.
57. Steve Pruitt and Tom Barrett, "Corporate Virtual Workplace," in Michael Benedikt, editor, *Cyberspace: First Steps* (Cambridge, Mass.: The MIT Press, 1991), 383–409.
58. Ellen Ullman, "Out of Time: Reflections on the Programming Life," in James Brook and Iain A. Boal, editors, *Resisting the Virtual Life: The Culture and Politics of Information* (San Francisco: City Lights, 1995), 131–132.
59. While new work designs promote intensive use of intellective skills, they require the new working body to be mostly immobile, reducing it to micro-movements of the wrist and fingers. This discipline of the body, however, has failed to attain complete success, largely because it is hard to achieve a static body without injuring it. Sedentary work has resulted in Repetitive Strain Injury (RSI) and Cumulative Trauma Disorder (CTD) – two umbrella terms that stand for various work-related injuries to the muscles, nerves, and tendoms of the upper limbs, including carpal tunnel syndrome, bursitis, tendonitis, tenosynovitis, frozen shoulder, and epicondylitis. RSI and CTD have registered an increase of nearly 1,246 percent from 1982 to 1992 paralleling the rise of information technologies in corporate restructuring, as mentioned in David R. Hayes, "Digital Palsy: RSI and Restructuring Capital," in *Resisting the Virtual Life: The Culture and Politics of Information*, James Brook and Iain A. Boal, editors (San Francisco: City Lights, 1995).
60. David Harvey, *The Condition of Postmodernity: An Enquiry into the Origins of Cultural Change* (Oxford, England; Cambridge, Mass.: Blackwell, 1989).
61. Marshall McLuhan, *Understanding Media: The Extensions of Man* (Cambridge: The MIT Press, 1995 (1964)).
62. C. Wright Mills, *White Collar: the American Middle Classes* (New York: Oxford University Press, 1951).
63. R. M. Kanter, "The Future of Bureaucracy and Hierarchy in Organizational Theory," in P. Bourdieu and J. Coleman, editors, *Social Theory for a Changing Society* (Boulder, Col.: Westview, 1991).
64. Michael J. Piore, "Review of the Handbook of Economic Sociology," *Journal of Economic Literature* 34/2 (1996): 741–756.
65. Polly Callaghan and Heidi I. Hartmann, *Contingent Work: A Chart Book on Part-Time and Temporary Employment* (Washington, D.C.: Economic Policy Institute, 1991).

66. Peter B. Doeringer, *Turbulence in the American Workplace* (New York: Oxford University Press, 1991).
67. Stanley Aronowitz and William Difazio, *The Jobless Future: Sci-Tech and the Dogma of Work* (Minneapolis: University of Minnesota Press, 1994).
68. Pelle Ehn, *Work-Oriented Design of Computer Artifacts* (Stockholm: Arbetslivscentrum, 1988).
69. Malcolm McCullough, *Abstracting Craft: The Practiced Digital Hand* (Cambridge, Mass.: MIT Press, 1996).
70. Richard Edwards, Michael Reich, and Thomas E. Weisskopf, *The Capitalist System: A Radical Analysis of American Society* (Englewood Cliffs, N.J.: Prentice-Hall, 1978).