Cooperative work, coordinative practices, and computational artifacts

Inaugural lecture
Department of Organization, Copenhagen Business School
9 June 2009

Kjeld Schmidt
Good afternoon,

I’m truly honored to be here. The Department of Organization is widely known for not only academic excellence but also for staying focused on issues of concern to the world of ordinary work and organization — when so much of sociology reminds me of the old song by the Kinks about ‘dedicated followers of fashion’.

It was therefore with a deep sense of humility that I received the news that the assessment committee found my research of sufficient relevance and quality and that the presidency of CBS decided to offer me the professorship.

I’ll today talk about some of the work I have been doing over the last couple of years. In fact, it’s about work in which I’m currently still engaged, so if it seems a bit rough, then that’s exactly how it is. There’s still a lot of work to be done. But then it’s my inaugural lecture, not my departing one.

I’ll start by talking about research in an area in which I’m a complete novice — but then you can’t ask questions or object, so I’ll take advantage of the rare opportunity to speak unopposed.

1. The accidental birth of a new technology

In 1936 something happened in mathematics that was to turn out to be quite important. In fact, it happened in an esoteric and abstruse corner of mathematics, namely, the attempt to ensure — establish for certain — that mathematics is consistent.

At that time, leading mathematicians had been deeply concerned with the foundations of their science. It was felt that the impeccable certainty and consistency of mathematical knowledge could be questioned. It was widely perceived as a foundations crisis. Mathematicians don’t take words like ‘crisis’ lightly, but then they felt evicted from Paradise. Several attempts were made to get back to that state. However, it turned out that the Paradise of impeccable certitude was not to be regained, but the sense of crisis has dissipated.

The story is of interest to us because mathematicians and logicians, in the course of trying to overcome the foundations crisis, developed some very sophisticated conceptual tools, calculi. What they developed — incidentally developed — became the conceptual foundation for computer technology.
In 1936 a young mathematician at Cambridge named Alan Turing had a paper published in the *Proceedings of the London Mathematical Society*. Its title was ‘On computable numbers with an application to the Entscheidungsproblem’ (Turing, 1936).

In his paper Turing attacked a problem that had been posed a few years earlier by one of the most reputable mathematicians at the time, David Hilbert, in a paper where he identified a series of problems that needed to be addressed to overcome the crisis. Since Hilbert was German the problem became known as the *Entscheidungsproblem*, that is, the problem of the decidability of mathematics; the problem being that of showing the existence of a definite method that could, in principle, be applied to any assertion and which was guaranteed to result in a proof or refutation of the assertion (Hilbert, 1927).

In trying to address this problem, Turing developed a calculus that was presented in the form of an imaginary calculating machine. This type of calculus is now known as a universal Turing machine. It was, of course, an imaginary machine, a thought experiment really, but the calculus forms the basis of modern computing. The unique feature is that the program controlling the behavior of the machine is treated in the same way as data, a collection of signs residing in memory.

When the war broke out, Turing’s research was interrupted, as he became involved in cracking the German ciphers. In this work he was involved in using a specialized electronic calculator, the Colossus, that hammered away at the encrypted message traffic between German Naval Headquarters and the fleet submarines in the Atlantic. After the war, he resumed his research on the theory of computing but building on his experiences from the war, and in 1947 he sketched the design of the first universal electronic computer (Turing, 1947). There had been calculating machines around for centuries, and during the war some very advanced calculating machines like the Colossus had been built. However, Turing’s design is fundamentally different in that it is a (practically) universal calculator, a computer, as we would call it today.
This idea — the idea of ‘the stored-program computer’ as it’s called — is one of the truly revolutionary inventions. That it was developed as an intellectual tool for solving an esoteric problem that occupied a small community of mathematicians engaged in regaining firm ground for their science is just one of those charming shifts that one sometimes finds in the history of science.

2. The age of the cheap machine

The importance of this idea to the world at large becomes evident if we — very briefly — consider the evolution of machinery from the onset of the Industrial Revolution and up to the present.

As pointed out by contemporary observers of the budding technology of machinery, such as Andrew Ure and Karl Marx, what was radically new in what was going on in the factory districts of England was not the use of external sources of power (animal, wind, water, or steam power) but the transfer of the control of transformation processes — carding, spinning, weaving, and so on — from the worker to the implement (Ure, 1835; Marx, 1867).
The machines of the industrial revolution — such as Robert’s Self-Acting Mule spinning machine from 1825 — were mechanical in the sense that the transmission of power and the control of movement were physically integrated. Power was transferred to the tool or the work piece by means of belts, cog-wheels, gear trains, camshafts, pack-and-pinion, and so on, and those very same parts at the same time also regulated the movements of the tool (controlled the speed, direction, etc.). To construct and modify machines required significant skill and effort, and indeed, the cost were such that the use of machinery was restricted to a few branches of industry, typically mass production (Hirschhorn, 1984).

This picture has changed radically with Turing’s computer design: the stored program architecture. It makes it highly economical to construct control systems that are not physically integrated with the power supply. One can in fact consider the computer as a universal control system: it can be made to incarnate any control function, be it a spinning machine, a machining center, a typesetter, or a jukebox.

Now, given the mythology of the digital, it’s necessary to keep in mind that a software program that has been launched and resides in the computer’s memory, in RAM, is still a machine. It is just a physical as the Self Acting Mule. It is just as physical, it is just not tangible: you can’t touch it.

However, it is infinitely faster because the mass of the electron is many magnitudes smaller than a cogwheel, a camshaft, a crank, etc. It can move at a velocity close to the speed of light. What is equally important, is that software machines can be constructed more or less automatically. When the blueprint has been designed, that is, the source code has been written and tested, the code can be compiled and executed automatically. The costs of modifying a software machine like a spreadsheet model of a budget are insignificant compared to the cost of modifying, say, the gearbox of a car. And as soon as the software machine has been built it can be copied and distributed at an insignificant cost. And more than that, software machines can be linked: they can transfer data to between themselves, and one machine can trigger the execution of another machine, perhaps at another location. In this way, vast machine systems are being built. In fact, the Internet itself is a vast machine system that facilitates the construction and operation of other specialized machine systems.
The economic, organizational, and social consequences of this radical reduction of the cost of producing and modifying machinery are enormous, to say the least: we live in the midst of the turmoil unleashed by this.

Whereas machinery, until a few decades ago, was rare outside of mass production, it now becoming ubiquitous: from CNC machines and CAD to CT scanners and GPS navigation. Modes of working, working with machine systems, that were restricted to classical industry, are now, in important ways, becoming characteristic of medical work, movie production, scientific laboratories. The post-industrial society is industrial, through-and-through.

What I want to talk about today is one of the research issues that have arisen now that the vast machine systems can be constructed and deployed and even modified at insignificant costs.

With networked computers it is technically and economically feasible to build machine systems that regulate the coordination of cooperative work activities:

- workflow management systems,
- production control systems,
- scheduling systems and group calendar systems,
- project management systems,
- document management systems,
- configuration management systems (in software engineering),
- medical record systems, etc.

Moreover, it is — in principle — technically and economically feasible for these coordination technologies to be designed in such a way that ordinary workers of whatever profession can design and redesign the rules according to which their work is coordinated by the machine system.

These potentials have only been realized sporadically. Why? Because our understanding of cooperative work and its coordination is deficient, vague, patchy.

This deficiency has motivated a research program in which sociological investigations play a novel role: Computer-Supported Cooperative Work.

This field should not be mistaken for the field of Computer-Mediated Communication that focuses on the social and organizational ‘effects’ and ‘impacts’ of the widespread uses of networked computer systems as media of communication: from email to instant messaging to Facebook. The CSCW research program does not address ‘effects’ and ‘impact’. CSCW researchers will have severe methodological reservations as to whether such questions can be investigated rigorously. But anyway, CSCW is not reactive with respect to technology, it is proactive. It
was formed as a field of research oriented towards the development of new technologies, not as a branch of technology assessment or futurology.

One more thing, to avoid misunderstanding. CSCW is not a branch of systems development either. When we talk about technology we are not talking about glittering gadgets, systems, things; but we are talking about technical knowledge, concepts, principles, and so on. — that is, the technical knowledge that enables us to construct and use gadgets and systems. So, the development of new technologies is first and foremost a conceptual endeavor.

What is special in CSCW is that ethnographic studies and other kinds of workplace studies play a central role in the development of the conceptual foundation of these new technologies.

3. Cooperative work

A fundamental issue for CSCW as a research field is this: How do we identify and delimit the very phenomenon we are studying. In fact, is it a researchable phenomenon at all?

Years ago the Confederation of Danish Employers published a small book entitled ‘Cooperation’. In the text the Confederation presented its position on the Cooperation Councils that were then being established in Danish enterprises in order to create a forum for handling everyday issues between employers and employees. What makes me recall it is that the book is illustrated by a series of photos of everyday situations where workers are engaged in carefully coordinated joint activities. The point of presenting these photos in this context was obviously to say that just as these workers adjust their individual activities with those of
their colleagues — in the same way employees and employers should conduct themselves in a cooperative spirit.

I mention this example because it, in a rather obvious way, illustrates the confusion that arises when relationships of a different category are mixed up.

This category mistake is especially acute in parts of organizational theory when relationships of different category are confounded, as when socio-economic arrangements defined in terms of relationships of ownership are confounded with contractual relationships, or with relations of causal interdependence, or relationships of obligation and accountability, identity, etc.

The term ‘organization’ is of course used in everyday discourse to refer to complexes of relationships of different category (ownership, economic appropriation, contract, causal interdependency, collective identity). It is used flexibly, without much confusion. In everyday discourse the ambiguity is of no concern, for the context normally helps us sort it out. Outside of everyday discourse, however, when organizational theorists try to generalize (as they should!), this often leads us to compare phenomena that are incompatible, for instance when ordinary relations of causal interdependence are matched with contractual relationships.

What the photos show are relationships of a different category than that of employee and employer. The photos show relationships of causal interdependence, whereas what the Confederation of Danish Employers was talking about were contractual relationships. That is, relationships among actors with partly diverging — and thus potentially conflictual — interests.

We do talk about cooperation in this sense too: ‘You are not being very cooperative now!’ — meaning: ‘Don’t insist so much on your own particular interest!’ This concept of cooperation is of a moral category: ‘We have all agreed to being part of this coalition, so let’s make it work.’

The concept of cooperative work is not of this category. It’s a technical category. The concept of ‘cooperative work’ was developed during the industrial revolution by theoreticians trying to express the increasing importance of work that systematically involves of the concerted effort of more than one worker.

The concept highlights a category of relationships characterized by causally interdependent activities, irrespective of the socio-economic framework of these activities, that is, irrespective of whether the work is voluntary work or wage work, neighborly help or slavery. In the same way, we can observe cooperative work relations that cut across boundaries of ownership, just as we can find people working next to each other, employed with the same firm, and yet not engaged in interdependent activities.

Now, with the intensification of the division of labor, the relationships of cooperative work tend to become increasingly complex. The coordination technologies that have been developing over the last few decades can be seen as so many measures to handle this increasingly complexity: workflow management systems, etc.
4. Coordinative practices

Enter a modern workplace, any modern workplace and look around. And you’ll see all kinds of linguistic artifacts: signs, boards, charts, routing schemes, handwritten notes.

These linguistic artifacts are the visual feature of some highly evolved practices — specialized practices that in different ways serve the purpose of maintaining order in complex work settings. We can call these sophisticated specialized practices *coordinative practices* and the artifacts in which they are expressed *coordinative artifacts*.

- Medical work
- Manufacturing work
- Architectural work
- Air traffic control

I’m not talking about office memos, technical reports, drawings, blue prints, etc. but about those mundane and insignificant inscriptions that for instance are used in maintaining order in collections of letters, reports, memos, etc. They are, so to speak *elements of form*, not content.

Coordinative artifacts are typically distinct objects: sheets of paper, plastic, or cardboard, etc. but today of course also digital ‘windows’ or ‘screens’. Such coordinative artifacts may move about, migrating from one worker to the next, as
forms or *kanban* cards do. Or they may stay put, as part of a framework for objects that do move about, as name tags on shelves.

Coordinative artifacts typically form a complex of artifacts that, as a collective, serves to maintain order.

These practices often include techniques that are devised by specialists, for instance planners, logistics engineers, etc. But they certainly also include techniques that ordinary workers devise for their own use. We see this wherever we enter a workplace. Learning to use them is part and parcel of the development of professional competencies — *but also learning to construct* them.

How can these practices be understood? When I claim that these artifacts serve the purpose of maintaining order, how do they do that?

At this point confusion reigns. The problem has its roots in the mechanistic account of rational behavior. When we act smartly, intelligently, and so on, our actions are determined — in a causal sense — by rules, plans, schemes, etc.

Against this mechanistic account, especially strong in cognitive science, an alternative account has been developed that maintains that rules, plans etc., do not determine plans in any strong sense; that action is essentially *ad hoc*.

Together these two positions have caused something like a case of intellectual constipation.

When we observe that actors typically take the next step as specified by the standard procedure and do so routinely, without reflection, without contemplating alternative routes of action, without bewilderment and struggling to make sense, then it is categorized as either the execution of a causal mechanism, or an artful accomplishment.

Neither account will help us to investigate how coordinative artifacts are actually used.

The source of the problem is that neither position recognizes that rules, plans, etc. are essentially *normative* constructs. The mechanistic position gives a causal account of normative action, whereas the alternative position states that this is wrong — and gives an account in which the normativity of constructs such as rules, plans etc. plays no part whatsoever.

To dissolve this hand-up, we should first of all resist the temptation to mystify the concept of *the causal* versus *the normative*. We should avoid any ontological interpretation of these concepts — in the style of the Cartesian mind-body gap — that leads us into endless and senseless speculation about how the normativity of ‘the social’ might ‘interface’ with the casualty of ‘the material’.

The fact of the matter is that these concepts belong to different domains of discourse (Pitkin, 1972), to what Wittgenstein calls different ‘language games’ (Wittgenstein, 1945-49). We can talk about a particular event in terms of intentions, rules, conventions, that is, in normative terms. And we can talk about *the same event* in terms of what made it come about, what led to this, — that is, in causal terms. For example, when judging whether a homicide was ‘murder’ or just ‘manslaughter’, or whether the assailant was insane at the time and hence innocent.
These distinctions are relevant to proceedings at a court of law but also in daily life: ‘I’m sorry I’m late: I was involved in a train accident.’

That is, we tend to talk about human activities in terms of normativity when issues of responsibility and justification are involved — whereas we tend to account for the same events in terms of causality when responsibility and justification are irrelevant. For instance, when accounting for the migrations of tribes from Central Asia to Europe some 1,700 years ago, the explanations one would consider would be something like drought in Central Asia, pressure from overpopulation, or whatever. Here justification would normally be quite irrelevant.

Similarly, we will routinely refer to the ‘unintended consequences’ of certain actions and measures, to ‘the fog of war’, to the ‘invisible hand’, etc. — thereby mobilizing a causal category of explanation, in contrast explanation in terms of justification.

These different domains of discourse, or language games, are categorically distinct. That means: we can not conflate a causal and a normative account without talking nonsense. We can talk about the same occurrence in both causal and normative terms, but not in the same discourse. For instance, when somebody says that the acts of a particular insurgent movement is to be explained with reference to poverty, repression, injustice, then that is not an explanation in terms of justification. It’s a causal account.

When we follow a rule, our actions are not caused by the rule. Or rather, a causal account of following a rule would, for example, be a description of how we were taught to apply this rule, how we were trained, etc. But when we follow a rule, the rule provides a guide, a criterion of correctness. Similarly, a plan, a procedure, a workflow specification, does not cause us to act in a certain way. We let our actions be guided by the plan simply because the plan gives us the criterion of what is correct conduct or the correct result under the given circumstances and because we have learned to master the appropriate techniques. In our doing so, the plan plays a normative role.

Now, what happens when rules, plans, procedures, schemes are — so to speak — mechanized?

5. Computational artifacts

I now come back to where this talk began: with the notion of computation and machinery.

What is a computational artifact?

It so happens that Wittgenstein subjected this kind of question to critical scrutiny. In fact, his many comments on this issue are, to a large extent, a critique of the way in which Turing and other mathematicians presented their findings — not their findings as expressed in mathematical propositions, but in their philosophical interpretations of their findings.

To suggest an answer to that he sketches some thought experiments. An updated version could sound like this one:

Imagine that somebody, say, an ecologist, is engaged in investigating a habitat in Central Africa. He is using a laptop computer and is busy entering data he has collected in a spreadsheet in order to produce some diagrams. However, for some reason he leaves abruptly and forgets to bring his laptop. Now, a group of chimpanzees comes along and finds it. They touch the keyboard and it wakes from sleep and resumes its operations.

Do the chimpanzees now calculate? We would not say so. A bathtub is only a bathtub to a life form that takes bath. A frog swimming in the bathtub does not take a bath. That is, ‘to calculate’ involves more than moving numerals about, it involves following rules and applying criteria of correctness.

A computer is a computer only as part of a practice in which people is trained to use it, in which it is routinely applied, and so on. Outside of that its just plastic, aluminium, copper, and tiny pieces of silicon. A pile of junk, really.

A note of clarification is called for here. It is clear that when computer scientists are engaged in programming, discussing programs, debating the complexity of an algorithm, — then such usage is of course completely legitimate. It’s simply a kind of shorthand. It’s comparable to using expressions like ‘sunrise’ and ‘sunset’. The problem arises when computer scientists (and journalists) bring these professional metaphors into ordinary language. Then a bonfire of metaphysics erupts.

Now, computation by means of electronic computers is certainly different from computation by means of of paper and pen, by means of using slide rulers or abacuses. Computational artifacts are machines: they operate more or less automatically. That is to say, they undergo state changes without human intervention (at least for a period of time).

This makes it a brilliant technology. But it is still a technique of computation by virtue of being used by practitioners in their normatively defined practices.

What does this mean when it comes to computational regulation of cooperative work?

What is new is that we are beginning to construct devices that we use in our coordinative practices — to regulate our interactions.
Probably the first example of this kind of technique is the mechanical clock in the town hall tower that regulated the life of the town. In the words of Lewis Mumford:

‘the regular striking of the bells brought a new regularity into the life of the workman and the merchant.
The bells of the clock tower almost defined urban existence’ (Mumford, 1934).

Now, the town hall clock represents a very simple coordination technology: it has two states: it either sounds or it doesn’t. More sophisticated designs had the number of strikes correspond to the time of day.

The coordination technologies we are now faced with are more sophisticated:
• a calendar system may issue notifications of up-coming meetings.
• the system may calculate and suggest possible times of planned meetings, etc.
• workflow management systems are used for routing documents, case folders, etc, through a system of division of labor, according to case types, formal roles, etc.,
• such a system may be used also for keeping track of deadlines.

The point I want to make is it that the problem with computational artifacts is not a principled one, but a problem of cost — the cost of construction and modification of such artifacts.

With electronic computers (with high-level programming languages and the rest) the construction of machinery has become immensely inexpensive. The cost of modifying such machines is negligible compared to previous technologies. It has over the last couple of decades become economically feasible to construct vast machine systems. And it is becoming economically feasible for ordinary workers to devise, adopt, modify computational artifacts that can play the role of a technique of regulation together with other techniques.

Like the town hall clock, the computational artifact does not cause practitioners to behave in a certain way. Like the medieval townspeople would let their activities guide by the hands of the clock, or by the sound of its bell, workers today can let computational artifacts guide their interactions, by providing agreed-to criteria for what is correct under given circumstances. Or like the medieval townspeople they can break the rule and decide that they have reasons — valid reasons — not to do so.

The problem is not conceptual. It is not principled. It is practical. The challenge is devise technologies that allow ordinary workers to construct and modify such coordination machines and to do so with minimal effort.

To do this, however, requires that we understand how coordinative artifacts are being constructed, negotiated, adopted, amended, combined and recombined. Is
there a pattern, a logic even, to the way they are constructed, put together, combined and recombined, and so on? To address those questions systematically requires that we get rid of our intellectual constipation.

6. Bibliography


