Impacts of Machine Learning on Work Design

Kevin Crowston
Syracuse University
School of Information Studies
crowston@syr.edu

Francesco Bolici
Università degli Studi di Cassino
Department of Economy and Law
f.bolici@unicas.it

Abstract

The increased pervasiveness of technological advancements in automation makes it urgent to address the question of how work is changing in response. Focusing on applications of machine learning (ML) that automate information tasks, we present a simple framework for identifying the impacts of an automated system on a task. From an analysis of popular press articles about ML, we develop 3 patterns for the use of ML—decision support, blended decision making and complete automation—with implications for the kinds of tasks and systems. We further consider how automation of one task might have implications for other interdependent tasks and how automation applies to coordination mechanisms. Our main conclusion is that designers have a range of options for systems and that automation of tasks is not the same as automation of work.

Keywords: work design, automation, machine learning, artificial intelligence

Impacts of Machine Learning on Work Design

Introduction

The evolution of work design—long interlinked to technology—has recently been accelerated by the increased capabilities of artificial intelligence (AI), machine learning (ML) in particular. ML can support the automation of a broad range of activities, including many decision-making tasks that until recently were the exclusive domain of humans. For example, ML is being applied to tasks ranging from credit-card fraud detection (Chan & Stolfo, 1998), to detecting skin cancers (Esteva et al., 2017), to advising in judicial decisions (Berk & Hyatt, 2015). Because ML-based systems can handle a greater range of decision-making tasks, new relationships are possible between workers and automated systems leading to new work designs.

Much of the rhetoric around work and AI focuses on people being replaced by automated systems. However, this view of the relationship between people and machines is too simplistic, because automatable tasks rarely stand in isolation (Chui, Manyika, & Miremadi, 2015). As a result, analysts expect that "technological disruptions such as robotics and machine learning—rather than completely replacing existing occupations and job categories—are likely to substitute specific tasks previously carried out as part of these jobs" (World Economic Forum, 2016). But the impact of such partial automation needs to be explored.

For instance, consider the work of a "computer user support specialist", a job we will use as a running example in this paper. It may soon be (if it is not already) feasible to develop an automated system to answer computer users' support questions (Masongsong & Damian, 2016). However, to be functional, such a system needs to fit the complex work of an organization. Someone must identify that there is a problem, collect relevant information to input to the system, explain the diagnosis to the user, implement the fix and so on. All of this surrounding work needs to adapt to an automated computer support system (and vice versa). The research question we address in this paper is: what are the implications of different patterns of relationship between human and ML-based automated systems for work design? We first develop a simple conceptual model of task automation and use it to identify different patterns of relationships between humans and machines, using illustrations from bellwether settings drawn from published reports. We then explore the implications of the distinctive nature of ML systems for its application in each pattern. Knowing the answer to our research question is important to expand our conversations about AI beyond an all-or-nothing focus on automation and work and to encourage thinking about alternative ways to deploy systems and to improve work design.

Theory

In this section, we develop a conceptual model of task automation to analyze how ML-based systems might have an impact on the design of human jobs. We first present a model for analyzing jobs, then discuss novel features of ML, then combine these perspectives to develop a set of issues to consider while analyzing the relationship between work design and technology.

Work design

We start by presenting our perspective on human work. In their jobs, most workers do a variety of different actions that might be more or less susceptible to automation. As noted above, a job is therefore not the right level at which to understand the impacts of technology. We follow the job analysis approach (United States. Employment and Training Administration and U. S. Department of Labor, 1991) in considering a job “an aggregation of tasks assigned to a worker” (Wong & Campion, 1991). In turn, a “task represents certain processes in which the worker, through his or her actions, transforms inputs into outputs meaningful to the goals of the job by using tools, equipment, or work aids” (Wong & Campion, 1991). The Employment and Training Administration of the U.S. Department of Labor has a database called O*Net that provides detailed information about jobs, including the comprised tasks. For example, the top three tasks (of 16) given for a “computer user support specialist”[1] are:

1. Answer user inquiries regarding computer software or hardware operation to resolve problems.
2. Oversee the daily performance of computer systems.
3. Read technical manuals, confer with users, or conduct computer diagnostics to investigate and resolve problems or to provide technical assistance and support.

In summary, the design of work is defined as “the content and organization of one’s work tasks, activities, relationship and responsibilities” (Parker, 2014).

Automating tasks

We next consider how a new technology might enable a task to be executed by a worker in a different way or to be completely automated. By automation, we mean the capability of a system to perform some task without human involvement. Since we are focusing in

[1]https://www.onetonline.org/link/details/15-1151.00. A very similar list of tasks is defined by the ESCO for the European Union at http://data.europa.eu/esco/isco/C3512

this paper on the applications of ML, we restrict our analysis to information-processing tasks, i.e., we do not consider the impact of robots on physical work.

For this analysis, we apply a model from Parasuraman, Sheridan, & Wickens (2000) that suggests decomposing information-processing tasks into a "simple four-stage view of human information processing": 1) information acquisition; 2) information analysis; 3) decision and action selection; and 4) action implementation (see Figure 1). Essentially, the steps of information analysis and decision and action selection match a particular action to a given set of input conditions. For each step, we consider if it can be partly or fully automated, meaning that the particular step can be done by a system without human intervention. Given this model of a task, we characterize tasks along the dimensions of inputs, outputs and the mapping between them.

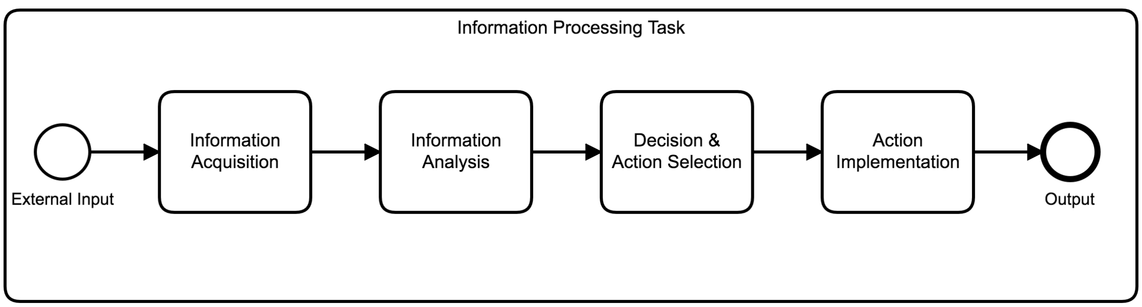

Figure 1. A simple four-stage model of human information processing for a task.

Inputs are the information acquired for the task. Drawing on the definition proposed for big data, we characterize these inputs by the volume, velocity and variety (Laney, 2001) of the information acquired. For example, some tasks like answering user computer support queries might have a high volume of requests in total, arriving at a high rate during certain times of the day (velocity) with a high variety of different queries, some more common, but with lots of exceptions.

Outputs can also be characterized by the 3Vs. In this case, by variety we consider a number of possible actions to be selected among. The decision could be binary (e.g., parole/no parole for a judicial decision or cancer/no cancer for a radiological screening) or of very high dimensionality (e.g., hundreds of possible replies in a customer-support setting or for a more complicated medical diagnosis). Again, we also need to consider the distribution of the outputs, whether some outputs are more common than others, i.e., the proportion of exceptions (Perrow, 1967).

Finally, we consider the complexity of the decision rules that connect inputs and outputs, which covers the steps of information analysis and decision and action selection. These rules could be very regular (i.e., high analyzability (Perrow, 1967)) or very irregular (low analyzability).

The above discussion has considered inputs, outputs and the mapping as static, but there could also be a dynamic aspect. For example, the nature of the inputs and outputs could change over time rather than being static and pre-given. Tasks are most likely repeated, so the information acquired as inputs could include feedback from prior rounds. And the mapping rules could evolve as system learns or as inputs and outputs change.

Given this model, we can consider what parts of the task-framework are automated. We have suggested that information acquisition has to be at least partly automated for the task to be automatable, but performance of a task might also rely on information held by humans. Similarly, information analysis and decision selection could be done by humans, automation or some mix, as is the case for action implementation. These combinations yield a number of patterns of automation. Pacaux et al. (2011) identified 10 levels, some with sublevels. Level 1 is no automation, i.e., all four steps performed by a human, while level 10 is total automation, i.e., all four steps performed by a machine without human intervention. As we assume that information acquisition and decision implementation are at least partly automated, we consider a subset of the 10 levels, which will be presented below.

Technical drivers for automation

Given this framework, automation is appropriate, and indeed likely has already happened, for tasks with a high volume of data, arriving with high velocity and where there is a high need for replicability of the decision making. For example, credit card fraud detection has to be automated because of the volume of transactions and the need to make the decision quickly and accurately. Similarly, stock trading is largely automated.

We recognize two technological drivers that support an increase in automation. The first driver is digitization: increasingly more data and interactions are digital. The greater penetration of digitized data implies that the data acquisition and decision implementation steps of our task model are done via a system, increasing the range of tasks executable through machine processing. For example, consider our computer user support specialist answering user inquires. These inquires could be made face-to-face or over the phone. However, if they are submitted via a computer system (digitally), then they can be processed by the system, opening the potential for automation. Similarly, if the users' computer systems are networked, an automated system could act on them directly to address problems, expanding automation to include decision implementation.

The second issue is about what can be automated in the intervening steps of information analysis and solution/action selection. Situations with very irregular rules—i.e., low analyzability (Perrow, 1967)—resist automation due to the variety of data and exceptions to the connection between input and output. What recently changed is that the capabilities of automated systems have improved. In the past, decision-making was automated with a set of rules: if some parameter or combination of parameters have particular values, then a particular decision is taken. ML system provide new capabilities for complex pattern recognition. Rather than having to make explicit ”if-then” rules, a system can learn the appropriate outputs given a large set of training examples (input-output pairs). For example, given a sufficient volume of known fraudulent and legitimate transactions, a system could learn what combinations of a transaction's characteristics suggest a fraudulent activity. Furthermore, a system can learn from cases over time and so continue to improve. Having been trained, an ML-system given some entity will give a score for how well it matches the pattern. The score can be converted to a decision by comparing it to a threshold, fixed or compared to others in a pool. Applying ML, a system can learn to identify solutions that were not coded ex-ante by humans and thus handle less analyzable mappings between inputs and outputs. Even where tasks are already automated, automation can be improved by refining the quality of the mapping from inputs to outputs.

However, ML systems have distinctive characteristics that are unlike prior systems for supporting or automating work. A first major difference is that ML performance depends heavily on the quantity and quality of data available for the training. ML typically requires a large training dataset to learn from, with high volume of inputs and outputs. Furthermore, as systems are reliant on data, they often exhibit hybrid agency, combining human and machine actions; human to generate an initial dataset and then further ML-based actions, meaning that initial human biases may be amplified. Second, the results of ML are most often probabilistic: e.g., when classifying an unknown case, an ML system likely provides probabilities that the unknown case fits one of the known categories rather than a definitive answer. Finally, many ML techniques are opaque: unable to explain why a particular output was selected.

Overall, ML systems behave quite differently than programmed systems. These differences can cause problems for use and users. The application of an ML system is clearly an algorithmic phenomenon, but our ability to control the technology is limited: an unwanted behavior is hard to fix if it is the result of training rather than programming and the precise reason for the answer can not be easily pinpointed. For example, engineers at Google were embarrassed when their image-labeling-system labeled a black user as a “gorilla”, but reportedly the only solution so far has been to eliminate the term “gorilla” from the labels (Simonite, 2018).

We suggested above that the trends towards digitization and the increased capabilities of ML-systems point towards increased possibilities for automation. However, the possibilities for automation depend on nature of the task, particularly the proportion of exceptions in the inputs and outputs and the stability and analyzability of the mapping between them (Perrow, 1967). Stable, routine tasks, those with high analyzability and few exceptions, have little or no need for information analysis or decision and action selection, meaning that the worker can just implement the actions. Such tasks are also very automatable.

If the task has low analyzability, but few exceptions, then analysis is hard, but the selection of actions is from limited range. These tasks may be increasing amenable to automation with the capability of ML-systems to learn patterns. For tasks with high analyzability but many exceptions, analysis may be easy, suggesting automation, but large number of choices for action may be problematic for ML, both in ensuring that the training data are complete and for achieving the necessary precision. Non-routine tasks are both low in analyzability and high in exceptions, suggesting that automation will be difficult. And finally, unstable tasks, ones for which the inputs, outputs or the mapping evolve over time, will also be challenging to automate.

Task interdependencies

The analysis above has considered an individual task. But jobs are collections of tasks, not just one, and furthermore, people doing a job typically have to interact with others. As a result, the impact of using ML for a task will propagate beyond the boundaries of the task itself.

To analyze multiple tasks, we consider how a particular task is interdependent with others, defined as “the extent to which the inputs, processes, or outputs of the tasks affect or depend on the inputs, processes, or outputs of other tasks within the same job” (Wong & Campion, 1991). For example, the second task in the list for a computer user support specialist is to monitor system performance. It may be that handling problem reports from users is helpful to see when a system has changed, because the kinds of problems change. If that first task is entirely automated, the specialists will need to develop new ways to get information about the systems.

We can also consider interdependencies between tasks that compose different jobs. An isolated task might be automated with few consequences, while one that interacts with many other jobs will be more problematic. While this perspective is quite common in studies of organizational design, it is interesting to note that the O*Net database does not explicitly record task interdependencies or what other jobs a job interacts with.

To analyze interdependencies, we adopt a coordination theory approach (Crowston & Osborn, 2003; Malone & Crowston, 1994). Malone and Crowston (1994) analyzed group action in terms of actors performing interdependent tasks to achieve some goal. These tasks might require or create various resources. The actors face coordination problems arising from dependencies that constrain how tasks can be performed.

The key point in coordination theory is that the dependencies create problems (or possible synergies) that may require additional work to manage. The necessary tasks of managing dependencies are what Malone and Crowston (1994) called coordination mechanisms. As the pattern of dependencies among tasks changes, we expect to see corresponding shifts in the needed coordination mechanisms.

In coordination theory, dependencies are conceptualized as arising because of the use of common resources among tasks. These dependencies come in three kinds. First, a shared-input dependency emerges among activities that use a common resource (like Thompson's pooled dependency (Thompson, 1967)). For these, the resource must be allocated to a particular user (if it is not shareable), e.g., through a schedule or first-come-first-served. If we consider as a resource the actor, either a human or a machine, who carries out the steps in the tasks, we can think of the resource assignment coordination mechanism instead as a task assignment mechanism that identifies which actor should work on which task.

Second, producer-consumer or flow dependencies match Thompson's sequential dependency (Thompson, 1967): one task produces a resource that a second uses. Flow dependencies including three sub-dependencies: the need to manage the usability of the resource as well as the timing and location of its availability. Considering usability, we might consider whether the producer of the resource adapts to the needs of the user or vice versa. For timing, we might consider whether the producer tells the user when to work or vice versa.

Finally, a shared-output or fit dependence occurs when two activities collaborate in the creation of an output (in the case where the output is identical, there is potential synergy, since the duplicate work can be avoided). The integration of different outputs might be done by in a variety of ways.

A final possible relation between two tasks is when one is a subtask of the other, that is, when the work to accomplish some goal is decomposed into smaller tasks to be performed. From a coordination theory perspective, the additional work needed to identify which subtasks to perform (i.e., planning) is another kind of coordination mechanism.

Impacts of automation

Decisions made about work design will have implications for the nature of the jobs that are created. Studies have identified a wide set of factors that characterize work designs (e.g., level of autonomy, feedback, interdependence, skill variety, physical), diverse outcomes of work design for workers (e.g., attitudinal outcomes such as satisfaction or motivation, behavioral outcomes such as performance or turnover, cognitive outcomes such as learning or identify and well being outcomes, such as anxiety, stress or burnout) and finally, mechanisms that link work design and work outcomes. Taken together, the outcomes for the workers have implications for organizations (e.g., overall productivity, skill and training needs, costs). More recent work has expanded work design to consider the impact of group work design and group outcomes as well (e.g., Morgeson & Humphrey, 2008).

A particular concern with automation is the ability of those interacting with the automated systems to understand what the systems are doing and to intervene if needed. For instance, there have been many studies of pilots interacting with intelligent autopilot systems that largely automate the job of flying. One outcome of this work is the identification of the problem of automation surprises (Sarter, Woods, & Billings, 1997), when the human operator loses track of the state of the automated system and so is surprised by unexpected or inappropriate actions or has difficulty taking over in a crisis. The design of autopilots and of the work of pilots have been redesigned to address (at least in part) the problem of automation surprises. But automation surprises are bound to become more prevalent and arise in new settings as more people engage with automated technologies, with clashes between human and machine cognition.

A potentially larger use case is represented by automated cars. Passive fatigue, distraction and communication problems have been identified as crucial issues in several crashes. For example, the US National Transportation Safety Board's accident report on the crash of a partly-automated Tesla car states that:

> the probable cause of the Williston, Florida, crash was the truck driver's failure to yield the right of way to the car, combined with the car driver's inattention due to overreliance on vehicle automation, which resulted in the car driver's lack of reaction to the presence of the truck.[2]

Thus, planning for automation should also include the design of the activities that were traditionally embedded into human work (e.g., when human drives, their attention level is supposed to be high; while when is the computer to drive the car, specific tasks should be designed to stimulate human attention). Carsten and Martens (2019) propose for example a set of guidelines for improving human-machine interfaces to reach the required level of reciprocal understanding between human and machine, to tune the right

[2] https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR1702.pdf

amount of trust, to stimulate the appropriate level of attention, to minimize the surprises and to reduce human stress.

Some of these considerations (e.g. mutual understanding, automation surprises, trust calibration, level of attention) seem particular problematic for ML-based systems, as the precise reason for an answer often can not be pinpointed and cannot be defined *ex-ante*. As a result, the organizational arrangements around their use may be different than for conventional automation.

Findings about task automation

In this section, we analyze the possible impacts of different kinds of automation of individual tasks. We present 5 patterns for the performance of a task, 3 including automation, based on the framework presented above. For each, we consider the implications of the distinctive features of ML.

No automation

We start with two base cases: an entirely non-automated system (Pacaux et al.'s (2011) level 1) and a task with some technology support but not for analysis or decision and action selection (Pacaux et al.'s (2011) level (2)). Level 1 means a human performing user computer support in person, as shown in Figure 2. All the steps of the task are done by the human. Also shown is how this task relates to another task, in this case, the computer user's interrupted task.

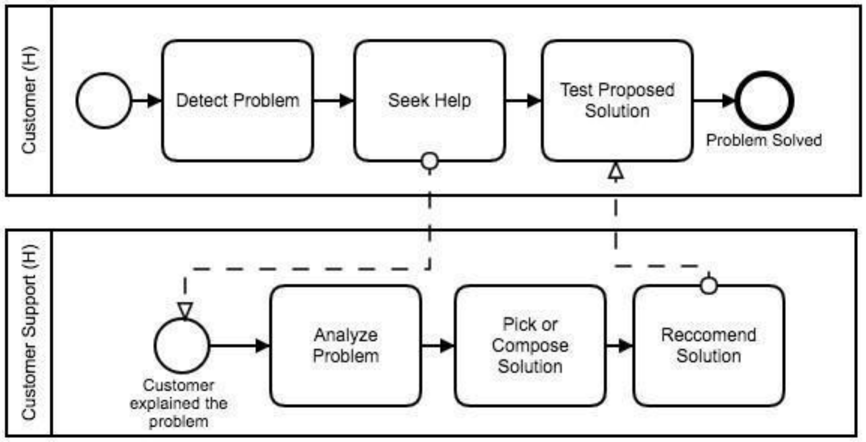

Figure 2. Level 1 automation: No technology support

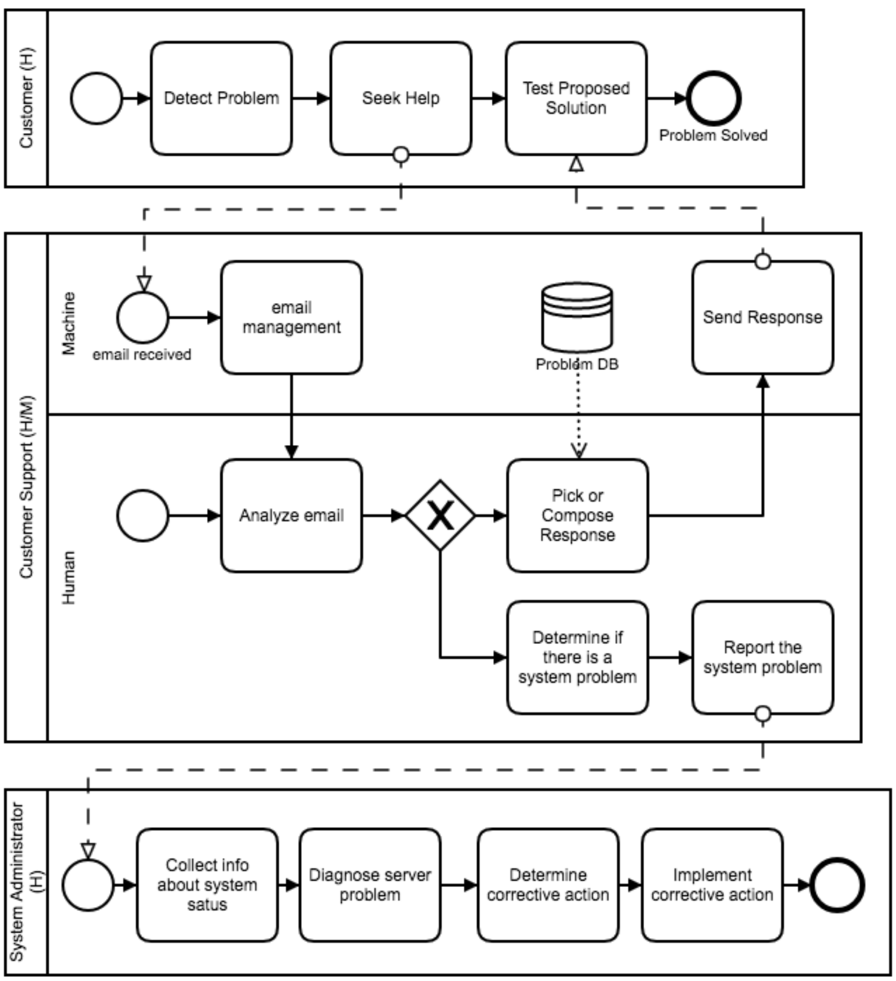

Figure 3. Level 2 automation: Digitization but not automation

A more likely scenario for user computer support involves some technology, even if there is no automation. For example, receiving and responding to problem reports via email adds digitization but not automation. In our framework, this change is represented by having just the first and last steps of information acquisition and action implementation supported by a system, as shown in Figure 3. In this case, a system provides information to support the human doing the task. Specifically, the person can collect information with system support (e.g., email or a problem reporting system) and take action through the system (e.g., send an email in response). We might also imagine support for other steps, e.g., a searchable database of problems to use in responding to a problem, but again, no automated steps. We acknowledge that this model is quite simplified. For example, even for computer system support, it is unlikely that the information needed will simply be provided. Instead, one of the skills of a computer support specialist is to be able to interview the user to determine the relevant details. Nevertheless, the model allows us to distinguish several patterns of automation.

Decision support

The first step in automation is that information analysis and action selection is supported for a subset of the cases encountered. We also assume the previous level of automation, that is that the information acquisition is digitized, so system has access to the data. In this pattern, the system provides decision support (Pacaux et al.'s (2011) level 3–4), meaning that the system can analyze the data and suggest one (level 4) or a few (level 3) responses, while leaving it up to the user to choose whether to accept and implement that response. This model is shown in Figure 4.

For example, in 2017 Google introduced a function called "Smart Reply", where the Gmail email system suggests three possible replies to a message, based on its analysis of a large, anonymized body of other emails. The user can choose to pick the suggested answer, to customize it or simply to ignore the system's recommendations (Kannan et al., 2016).

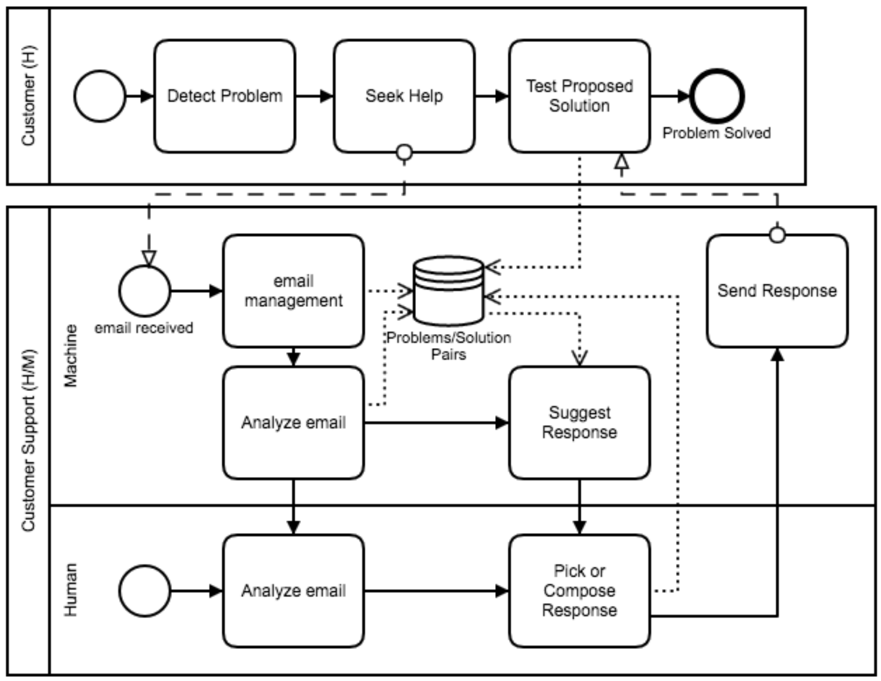

Figure 4. Level 3–4 automation: Decision support

A system could support salespersons by providing useful information to finalize a potential sale, but leaving it to the salesperson to decide if and how to use those suggestions. Udacity, an education company providing online courses, built a digital sales system to assist humans. When a potential customer asks a question, the system suggests an appropriate response that a salesperson can follow or customize. With this digital sale assistant, it is reported that the sales team was able to manage twice as many potential customers and to convert to actual customers a higher percentage of inquiries (Anthes, 2017).

NDA-Lynn is a system that can evaluate a non-disclosure agreement, identifying clauses that are too strict, unclear or not coherent with the whole agreement. Such a system could drastically reduce the time needed to read the document, allowing the lawyers to focus on reviewing the clauses that have been flagged as problematic by the system (Tauber, 2018).

In the US judicial system, some courts rely on ML systems to determine the probability that defendants will commit another crime or will appear for their court date. Kleinberg et al. (2017) suggest AI systems are able to improve trial accuracy by warning judges when a decision they are about to take (e.g., allowing or not allowing parole) is likely to be wrong. In China, the Judge Rui system can "automatically create and generate the judgment document according to the case" (Kee, 2019) that the judge can then edit and finalize.

Systems are also increasingly used by recruiting agencies to match companies and job-seekers. A company called Woo (Munford, 2017) developed a system to simplify the employment process, matching passive job-seekers' profile and skills to employers' needs. The system signals potential matches to the individual, who can choose whether or not to pursue the selected opportunity.

Many decision support systems have been designed for medical diagnosis. For example, a radiology system could provide warnings of possible anomalies in images to help separate urgent cases from those that can wait, while still requiring a human doctor's decision (Pearson, 2017).

The tasks in this category share some common features. A prerequisite is a high volume of input data with matched outputs to make feasible assembling a training set. Note that in some of these cases, the decision implementation step is not automated, meaning that the human both decides and carries out the decision. The locus of control stays with the human throughout the task and the human remains mostly embedded in the task rather than peripheral. The implementation of the system as decision support suggests that the human decision maker may have information that's not available to the machine.

We expect to see this pattern in cases where acquisition of information and decision implementation are mostly digitized, and where the input has high volume and velocity, so automation makes sense and is feasible, but the variety is high, so that pursuing complete automation is not feasible. Further, if the level of exceptions is high, identifying the relation between inputs and outputs is difficult, so human input is still needed.

Even though the system in the end defers to the human, implementation may raise some concerns. First is auditability: how does human know the system made the right recommendation? Will the human worker be able to recognize when to ignore the system? This issue is particularly problematic with ML systems that may be opaque and unable to explain the reasons for a decision. Second is deskilling: i.e., does relying on the system reduce the skill of the human in the long run?

While some of these concerns also apply to traditional Decision Support Systems (DSS), in ML their resolution appears to be even more complicated because, as Kusiak (1987, p. 2) stated, "the differences between DSS and AI systems is in the degree of intelligence". In traditional DSS the variables (and weights) taken into considerations to take a decision are determined *ex-ante* (through a what-if logic) and the outcomes are chosen in a defined interval of options. Instead in ML systems, the relationship among variables is defined emergently during the decision process and the outcomes themselves are most often probabilistic. Moreover the user control over the whole ML system is often limited, as well as the possibility to understand why a particular decision was taken, because the system is not transparent.

Blended decision making

The next step in automation presumes that for some set of the cases the machine can take and implement decisions by itself, what Pacaux et al.'s (2011) called level 6, blended decision making. This model is shown in Figure 5. For example, for user computer support, the system would automatically answer requests for which it was confident of an answer (e.g., the easy cases), deferring others to the human.

An example from the judicial system is the plan proposed by the Moscow City Court to use a system to prepare judicial decisions on so-called "indisputable cases". Human judges would intervene in such trials only in case of complaints about system-supported decisions, enabling them to focus on more controversial cases (Sysoev, 2018).

Examining the process models, we note that we are adding a new step, assigning some tasks to a human and some to the machine. To do so, the system has to know what it can do, as well as doing it. An alternative approach for cases with low individual volume is to have the human to decide which cases the system should do, as shown in Figure 6. For example, an architect might decide which aspects of a building design are sufficiently routine to entrust to an automated design system, and which should be done by humans.

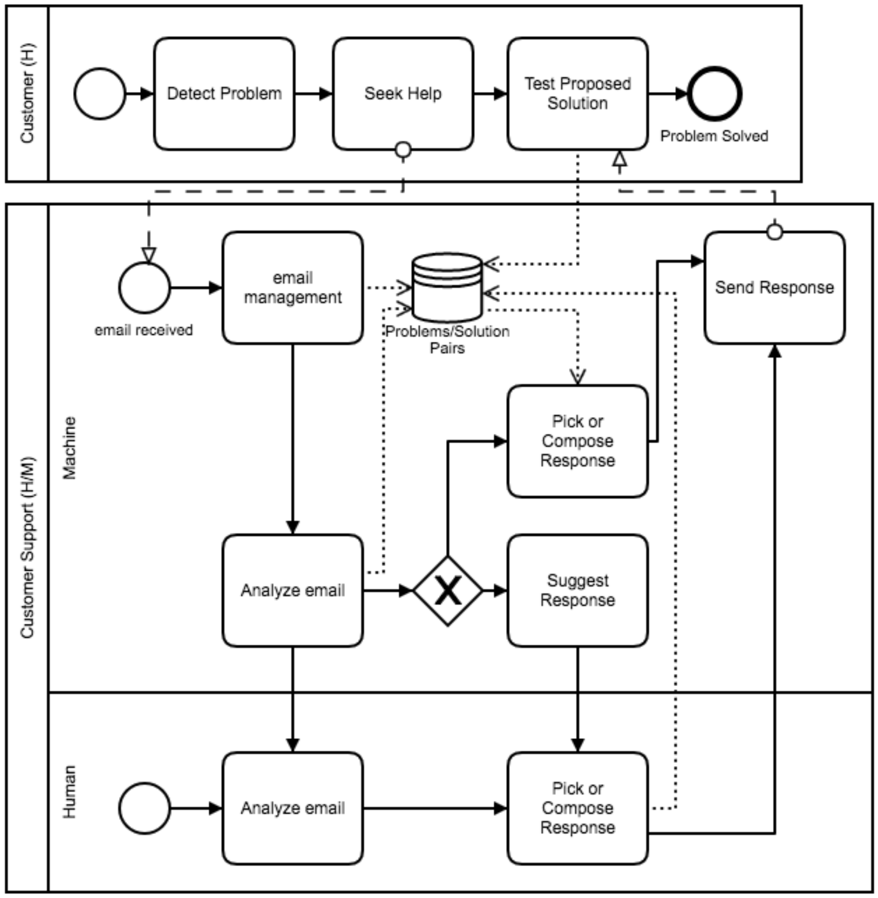

Figure 5. Level 6 automation: Blended decision making

For this pattern, again a large volume of training data will be needed. Furthermore, the system needs to be trained to recognize exceptions that should be deferred to the human. An interesting possibility in this case is that the ML algorithms can be periodically retrained on the cases performed by the human. In this way, the performance of the system can be improved, perhaps allowing it to take on more tasks.

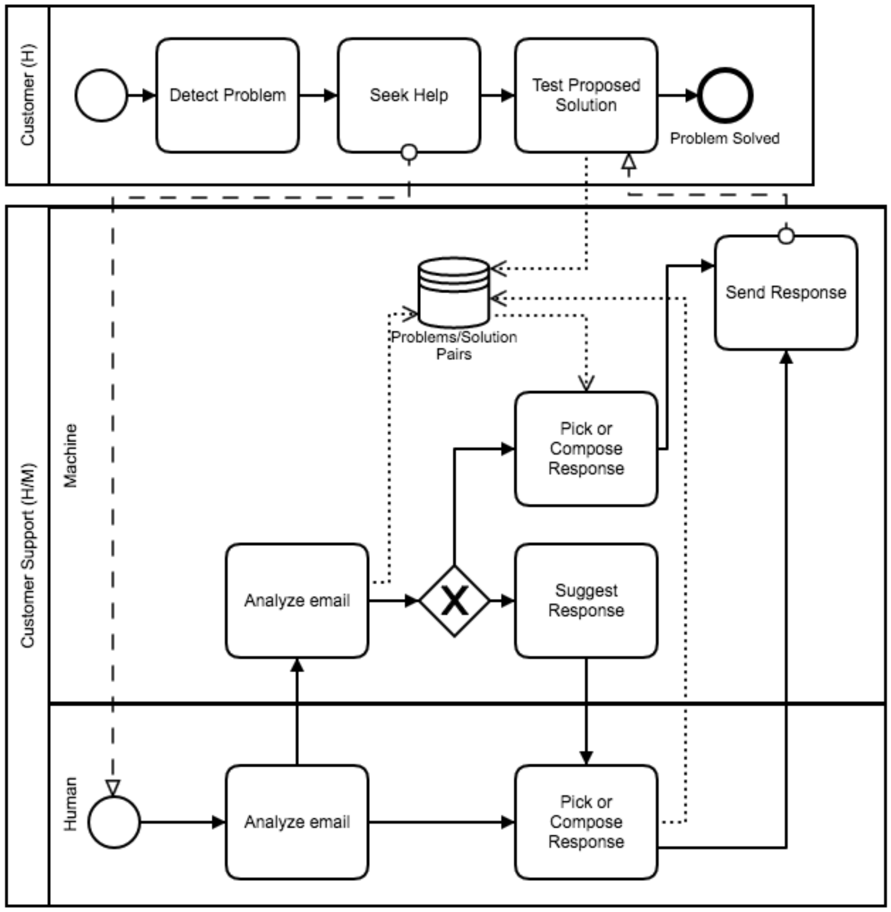

Figure 6. Level 6 automation: Blended decision making with human allocation of effort

The issues noted above for level 3–4 take on more urgency in this pattern because some actions will be implemented automatically. For example, how will human workers know what the system is doing and how will they be able to intervene if something goes wrong? Relying on customers to complain (as in the Moscow City Court case) seems problematic.

The question of deskilling takes on a different perspective in this pattern. It seems likely that the tasks the machine will do are the more straightforward one. A resulting concern is if those tasks constitute the entry-level job for the profession, e.g., the tasks that an architect delegates to the machine are those that would have been done by a junior architect. If these are instead done by the system, how does someone learn to do the task and enter the profession?

Task automation

Finally, we consider the case of complete automation (Pacaux et al.'s (2011) level 10; their levels 7–9 describe systems where the system seeks human input before implementing the decision). In this case, the system performs all of the steps above for all cases. If the implementation of the decision is automated, then the whole task can be done automatically, as shown in Figure 7. For example, computer user support might be provided by an automated system that parses a user's emailed report and automatically replies. In this pattern, the automated system does the task, while the role of the human is to monitor and possibly tune parameters to improve performance.

This pattern describes many already automated systems. In the stock market, fully automated trading systems are constantly buying and selling stocks on the base of their own predictive models and criteria. In Japan, Mizuho Financial Group recently introduced an algorithm-based trading service to predict how prices will change over an hour and find the best time to trade (Hyuga, 2017). Credit card companies broadly rely on anti-fraud systems for increasing the accuracy of automated and real-time approvals while reducing false declines (Cochrane, 2017).

We expect to see this pattern when the analyzability of the task is high and there are few exceptions (i.e., when the task is more routine), because the system has to handle all cases. As ML is applied, we expect systems to be able to handle tasks with lower analyzability, but not a high numbers of exceptions.

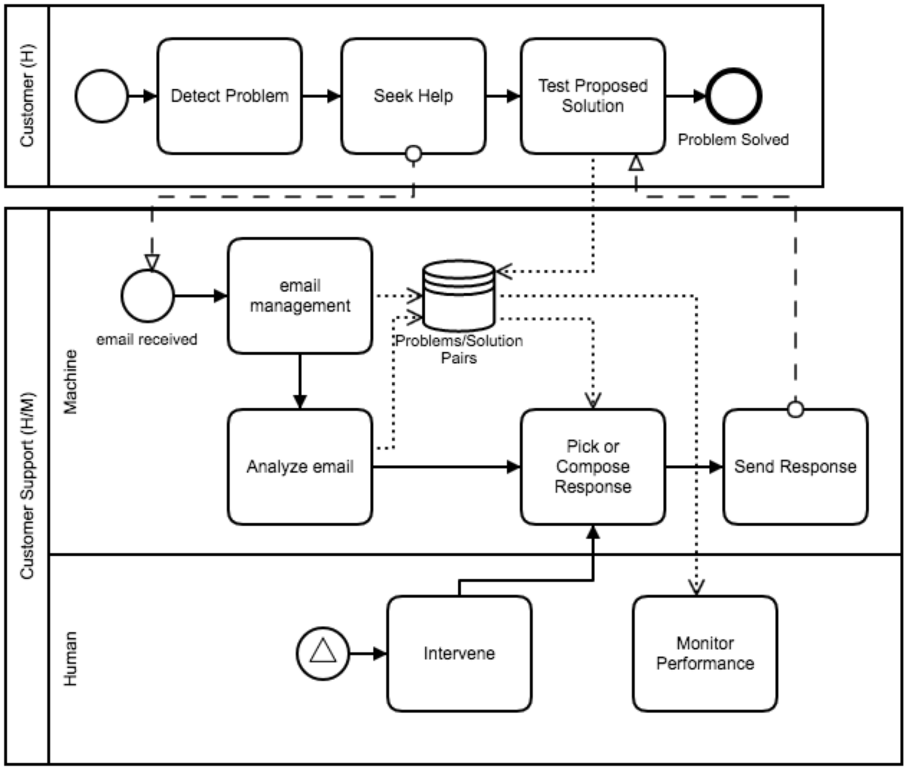

Figure 7. Level 10 automation: Complete automation with possible interventions or oversight

A question in designing such systems is the locus of control. Even if decision implementation is not automated and so is carried out by a person, a system could follow this pattern if the human is not given the authority to override the system by implementing a different decision. Alternately, even if the system is entirely automated, there may also be ways that a human can intervene in a particular decision for a specific case. For example, in social media users' reports of inappropriate behavior (or "bad" advertisements) are automatically checked by the system and appropriate measure are taken, however in specific cases, an editor can intervene to re-calibrate the automated answer.

As more tasks are automated, the role of humans in a system of tasks may reduce. In a study of bank automation, Adler (1986) described the process of peripheralization, in which automated processes relegate humans to the periphery of the processing: entering data into the system or monitoring its performance. Interestingly, peripheralization can have paradoxical effects. While automating a process might be expected to deskill workers because they no longer need to know how to perform the tasks in the process, Adler (1986) observed that it can also increase skill demands, as workers need to be able to comprehend the entire automated process to understand how their input affect the system and to debug problems.

As a result, this pattern poses the greatest risk for automation surprises. The model envisions a new role for humans as system monitors and meta-designers (i.e., designing the tasks for the machine to perform and setting the parameters), but how does human who's monitoring the system understand what it's doing and the impact of the parameters they can tweak?

Automation of multiple tasks

We now switch our consideration to the impact of ML-based systems across multiple tasks, within or across jobs. As discussed above, we adopt a coordination theory perspective to analyze dependencies among tasks and resources. This perspective leads us to consider two possible impacts. First, we consider how automating one task might impact another task with which it is interdependent. Second, we consider the implications of automating coordination mechanisms themselves.

Impact of automation on interdependent tasks

To illustrate the first situation, we consider issues when the output of one task is the input to the next (a flow dependency). For example, our computer system support specialist takes (or elicits) problem reports as input and produces recommended solutions as an output. The input and outputs are provided from and to some other task, as shown in the various figures above. As discussed in the theory section, flow dependencies imply three sub-dependencies: ensuring that the resource created is what is needed (also known as usability), that it is available when it is needed and that the resource is transferred to where it is needed. We analyze the impacts of automation by considering how these dependencies are affected.

Consider first automation of the upstream task. An example is an automated legal discovery system tasked with finding in a very large collection documents to be used by a lawyer (or team of lawyers) in assembling a pending legal case, as shown on the left of Figure 8. First, the usability sub-dependency means that the automated system needs to know what output is needed by downstream tasks. Coordination theory suggests approaches such as standardization (likely not applicable for discovery), asking the users what information they want or for the users to give feedback on the documents provided to refine the search. Furthermore, the results need to be presented in a usable format. For instance, a dump of thousands of documents is probably less useful than a directory organized by the reason for providing the documents, with appropriate meta-data.

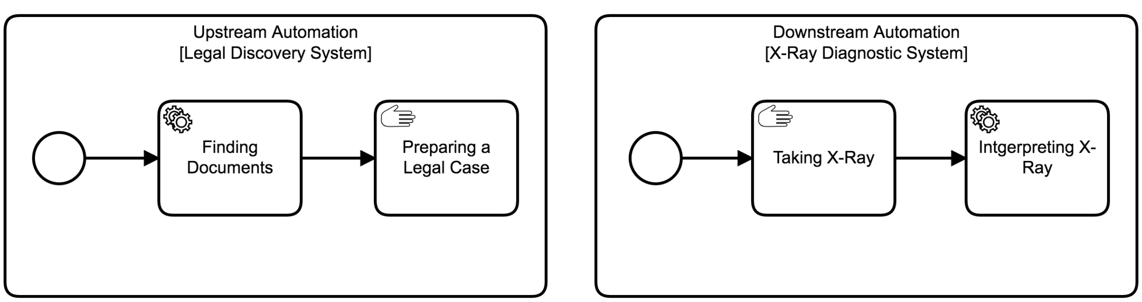

Figure 8. Flow dependencies between tasks when one is automated.

As for the second sub-dependency, timing, an automated system would likely be an improvement on a manual systems, as results could potentially be provided on demand, rather than being dependent on the availability of other workers. In other words, rather than waiting to be informed that documents are available, the lawyers could pull documents from the system. Similarly, documents may be easier to transfer (the third aspect) electronically than physically.

Contrariwise, the downstream task might be the one that is automated, for example, the task of taking an X-ray that feeds a task of interpreting the X-ray. If the later task is automated, as shown on the right side of Figure 8, then for usability it seems likely that the X-ray images provided would have to be standardized to meet the needs of the system. While standardizing X-rays seems feasible, the need for standardized inputs may be problematic if the information is hard to express, e.g., based on tacit knowledge. For example, the task of finding a house (part of the job of a real estate agents) seems automatable but in fact the decision can involve many emotional criteria that are hard for buyers to articulate. Part of the skill of an agent is eliciting these criteria. We noted above similarly that extracting useful problem reports is an important skill for a computer user support specialist. However, it is much more challenging to develop an automated system that can interact to elicit information from a user than to parse emails. Regarding timing, the second sub-dependency, automation might again be advantageous, as it might be possible to process the X-ray on demand, rather than waiting for a radiologist to be available. Finally, the transfer would have to be done electronically rather than physically.

This analysis can be extended to the next of tasks. For example, the X-ray diagnosis task has a downstream task, e.g., sending the patient for additional care. How the information about the diagnosis should be presented will depend on the needs of that task.

Finally, we note that the flows may be implicit rather than explicit. For example, as noted above, handling customer problem reports may be a way to learn about the status of systems, e.g., identifying problems by when the problem reports change. As a result, automating problem resolution may remove an important source of information. The opposite is also possible: if oversight is automated, the specialists may be unaware of ongoing system problems that affect users. Thus, there are cases in which the coordination is reached having visibility on other workspaces (and their changes). This combination of implicit and explicit mechanisms, also referred as stigmergic coordination (Bolici et al. 2016), often plays an under-appreciated role in supporting tasks alignment. Being emergent and sometimes informal, stigmergic coordination could be difficult to map while designing and implementing a ML-system, but to automate a task without considering its role as implicit coordination nexus could lead to a lack of communication flow into the organization and to the disconnection among some tasks.

Automation of coordination mechanisms

We next consider the possible implications of automating coordination mechanisms, which include coordinating shared inputs or outputs, coordinating flow dependencies and coordinating dependencies between a task and its sub-tasks.

Shared inputs

When tasks have shared inputs, meaning they use the same resources, it is necessary to have a task or resource assignment mechanism (i.e., a coordination mechanism) to decide which task gets which resource. Following the analysis we developed above, the task of resource assignment might be done manually, with decision support or entirely automated. For example, Uber assigns drivers to passenger trips using a fully automated algorithm, what has been referred to a “algorithmic management” (Mohlmann & Zalmanson, 2017).

In management studies, resource allocation is often studied together with the power balance and structure among different parts or individuals of the organizations. If automated resource allocation becomes more common it could result in a shift in power inside organizations or among workers. For example, Uber drivers perceive having little or no control over the assignments they receive or the implications of the system’s decisions for their pay (Mohlmann & Zalmanson, 2017).

Producer/consumer

The producer/consumer relationship represents the case in which the output of a task becomes the input of another task. As noted above, a producer/consumer dependency involves three sub-dependencies. In the prior section, we considered the implications for coordination when one of the tasks was automated. In this section, we discuss how the corresponding coordination mechanisms itself might be automated.

As mentioned above, an often overlooked interdependency is the fact that the produced item must be usable by the receiving activity. To assure usability we can rely on standardized solutions or the direct involvement of the users to explicitly express their needs. However, the coordination mechanism needed to determine usability can increasingly be automated thanks to the amount of available data, the possibility to integrate different sources and the velocity with which a certain output can be re-arranged according to the other variables. For examples, websites are getting “smarter”, personalizing their contents on the basis of past behavior and multiple user data points (e.g., time of day, where users are coming from, type of device they are accessing from, day of the week). Triangulating all these data can increase not only the usability perceived by the receiving actors, but even contribute to personalizing it according to their needs.

Considering timing, managing prerequisite dependencies often involve a notification process to signal the downstream activity that it can start. The activity of notifying, as well as sequencing and tracking, can be done by a person, can be supported or completed automated by a ML system. For example, there are AI home surveillance systems that, through face recognition, monitor people getting into the house, distinguishing those in the circle of people with access from those unknown that will set an alarm, sending a message to the house-owner or directly alerting the security company. These solutions can also be integrated with AI audio analysis systems able to to distinguish unexpected and unordinary sounds (e.g. a burglar trying to force the door) from usual background noise, in order to inform the owner that action is required.

Finally, considering transfer, with information products, it is easy for a system to manage the movement of an information product from one actor to another. Increasingly, it is possible to automate such movement for physical goods as well. For example in Amazon warehouses a combination of advanced computer vision and ML enables Amazon workers to grab an item coming from a distributor or supplier, scan it and place it in a bin. The system will recognize where the item was placed and record it for future reference.

Shared outputs

When tasks have shared outputs, there can be issues, e.g., with ensuring that the outputs are created at the same time or that they are coherent. Some of these activities can be automated, as for example using mobile apps like Pager or Carepy that promise to help to coordinate all the healthcare experts and resources that a patient need, ensuring that all the needed information are timely shared with and among the appropriate actors.

Task/subtask

The final category of dependencies is when a task needs to be decomposed into subtasks that accomplish the overall goal. Such planning is increasingly amenable to automation, e.g., using a model of a real world system as a basis for exploring the impacts of possible actions. For example, the Hong Kong Mass Transit Railway (MTR) relies on an AI system to schedule and manage all engineering works for its rail lines. The system proposes a plan that fits the various constraints and humans review the final plan (Chun et al., 2005), making it an example of decision support. Increasingly, the plans are implemented automatically.

Discussion

The patterns of automation discussed above differ in the level of automation, from individual support to process automation. Much of the rhetoric has focused on the latter case, but research is needed also on how to effectively do the former. As well, many tasks are performed by or with teams rather than solely by individuals, so research is needed to identify how a system can be an effective team member.

Despite the differences, the patterns have a number of commonalities. First is a need for a sufficient volume and quality of training materials and a sufficient regularity of the relationship between inputs and outputs. If the task has many exceptions, an ML might not be able to learn them, suggesting a decision support or blended decision making pattern. A particular challenge to the blended decision making pattern is the ability of the ML to know when it does not know and should defer the case to the human. Finally, if the task is not stable, automation will be challenging.

A second common issue has been transparency of decision making. In all of the models, there is a need for human workers to maintain awareness of the system's performance. As long as ML-systems do not provide full automation, the human understanding of the whole system is needed to ensure that: i. the agents (human and AI) collaborate in a safe manner, comprehending each other's intentions and actions; ii. the automation of certain activities would not obstruct the flow of implicit coordination mechanisms with other tasks; iii. humans would still pay attention to the tasks, even when not directly performing them.

A third issue as automation increases is to identify the circumstances under which is reasonable for the humans to be able to veto the machine. These interventions may decrease the reproducibility of the task (and so face managerial opposition), but they also acknowledge that the automated system may not have complete information. These two concerns (visibility and agency) are tightly coupled, because the former is necessary to be able to implement the latter. We note that a human worker may technically have the authority to make the final decision (i.e., the system nominally follows the decision support pattern) but face obstacles to exercising the authority, resulting in practice in complete automation. The pressure to follow the system could be from internal management or external forces. For example, a doctor who decides to ignore the advice of a medical expert system could risk a suit for malpractice for not following the encapsulated “best practice”. In such a situation, a doctor might feel forced to cede authority to the system.

A fourth issue, from a practice perspective, concerns the role of organization design in ML implementation projects. Shifting from an human to a ML task requires not only technological skills and competencies but also an organizational assessment of the nature of the task we are redesigning, with a specific analysis of its interdependencies – both the existing ones and those that will emerge with the automation. Implementing ML-systems means to redesign the organizational processes and to redefine the communication and coordination flow both at implicit and explicit level. Moreover applying ML to specific tasks would also lead to the design of other activities that were not needed before. For example this is the case of autonomous driven vehicles: as long as the driver is human there is a legitimate assumption that s/he will keep the attention up while performing the task, while with a autonomous vehicle there will be the need to design specific tasks (e.g., if the hands are not on the when set an alarm) to motivate the human-driver to pay attention to the driving itself.

Conclusion

The analysis of the articles in this paper was a pilot application of the conceptual framework. A shortcoming of relying on popular press articles for examples is that few discuss the impacts of using the system on workers in any detail. More detailed case studies will likely be needed to establish these connections. Finally, further analysis is needed of multiple intersecting tasks, not just pairs, e.g., when tasks are composed of multiple subtasks.

The main message of this paper is that, contrary to the rhetoric of an inevitable "march of automation", there are a variety of options for how automated systems can be used with differing impacts on jobs. The decision about which pattern to follow is partly driven by nature of tasks and system capabilities. But designers should resist technological determinism and be aware of the impacts of managerial decisions about how technologies should be deployed, especially about the desired locus of decision making. Designers should strive for a fit between system characteristics and the characteristics of the setting in which the automation is introduced. Our conceptual model can support designers in identifying different patterns of relationships between humans and machines, proposing a range of different scenarios for automation (not only the all-or-nothing cases) in which the deployment of the ML system is integrated with the work re-design.

References

Adler, P. (1986). New technologies, new skills. *California Management Review*, *29*(1), 9–28.

Anthes, E. (2017). The shape of work to come. *Nature News*, *550*(7676), 316.

Berk, R., & Hyatt, J. (2015). Machine learning forecasts of risk to inform sentencing decisions. *Federal Sentencing Reporter*, *27*(4), 222–228.

Bolici, F., Howison, J., & Crowston, K. (2016). Stigmergic coordination in FLOSS development teams: Integrating explicit and implicit mechanisms. Cognitive Systems Research, 38, 14-22.

Carsten, O., & Martens, M. H. (2019). How can humans understand their automated cars? HMI principles, problems and solutions. Cognition, Technology & Work, 21(1), 3-20.

Chan, P. K., & Stolfo, S. J. (1998). Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In *KDD* (Vol. 98, pp. 164–168).

Chui, M., Manyika, J., & Miremadi, M. (2015). Four fundamentals of workplace automation. *McKinsey Quarterly*, *29*(3), 1–9.

Cochrane, M. (2017). How MasterCard is using AI to improve the accuracy of its fraud protection. Retrieved from http://www.businessinsider.com/mastercard-artificial-intelligence-fraud-protection-2017-1?IR=T

Crowston, K., & Osborn, C. S. (2003). A coordination theory approach to process description and redesign. In T. W. Malone, K. Crowston, & G. Herman (Eds.), *Organizing Business Knowledge: The MIT Process Handbook*. Cambridge, MA: MIT Press.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115.

Hyuga, T. (2017). Mizuho is said to offer AI trading Before MiFID Overhaul. Retrieved from https://www.bloomberg.com /news/articles/2017-09-20/mizuho-is-said-to-offer-ai-trading-service-before-mifid-overhaul

Kannan, A., Kurach, K., Ravi, S., Kaufman, T., Miklos, B., Corrado, G., … Ramavajjala, V. (2016). Smart Reply: Automated Response Suggestion for Email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016).* Retrieved from https://arxiv.org/pdf/1606.04870v1.pdf

Kusiak, A. (1987). Artificial intelligence and operations research in flexible manufacturing systems. INFOR: Information Systems and Operational Research, 25(1), 2-12.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., & Mullainathan, S. (2017). Human decisions and machine predictions. *The Quarterly Journal of Economics*, *133*(1), 237–293.

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. *META Group Research Note*, *6*(70), 1.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *Computing Surveys*, *26*(1), 87–119.

Masongsong, R. P., & Damian, M. A. E. (2016). Help Desk Management System. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1).

Morgeson, F. P., & Humphrey, S. E. (2008). Job and team design: Toward a more integrative conceptualization of work design. In *Research in Personnel and Human Resources Management* (pp. 39–91).

Munford, M. (2017). Woo Launches The First AI-Powered Headhunter For Employers—She's Called Helena. Retrieved from https://www.forbes.com/sites/montymunford/2017/11/27 /woo-launches-the-first-ai-powered-headhunter-for- employers-shes-called-helena/#409adb9a67f8

Pacaux, M.-P., Debernard, S., Godin, A., Rajaonah, B., Anceaux, F., & Vanderhaegen, F. (2011). Levels of automation and human-machine cooperation: Application to human-robot interaction. In *Proceedings of the 18th World Congress of The International Federation of Automatic Control* (Vol. 44, pp. 6484–6492).

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, And Cybernetics—Part A: Systems and Humans*, *30*(3), 286–297.

Parker, S. K. (2014). Beyond motivation: Job and work design for development, health, ambidexterity, and more. *Annual Review of Psychology*, *65*, 661–691.

Pearson, D. (2017). Artificial Intelligence in Radiology: The Game-Changer on Everyone's Mind. Retrieved from https://www.radiologybusiness.com/topics/technology-management/artificial-intelligence-radiology-game-changer-everyones-mind

Perrow, C. (1967). A framework for the comparative analysis of organizations. *American Sociological Review*, *32*, 194–208.

Sarter, N. B., Woods, D. D., & Billings, C. E. (1997). Automation surprises. *Handbook of Human Factors and Ergonomics*, *2*, 1926–1943.

Simonite, T. (2018). *When It Comes to Gorillas, Google Photos Remains Blind*. Retrieved from https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/

Sysoev, G. (2018). Artificial intelligence will help Moscow judges make decisions. Retrieved from https://lenta.ru/news/2018/05 /29/judge/

Tauber, A. (2018). This AI checks NDAs for free: And offers a grim glimpse of the future. Retrieved from https://thenextweb.com/artificial-intelligence/2018/02 /27/ai-checks-ndas-free-offers-grim-glimpse-future/

Thompson, J. D. (1967). *Organizations in Action: Social Science Bases of Administrative Theory*. New York: McGraw-Hill.

United States. Employment and Training Administration and U. S. Department of Labor. (1991). *The Revised Handbook for Analyzing Jobs*. JIST Works Incorporated.

Wong, C.-S., & Campion, M. A. (1991). Development and test of a task level model of motivational job design. *Journal of Applied Psychology*, *76*(6), 825.

World Economic Forum. (2016). *The future of jobs: Employment, skills and workforce* *strategy for the fourth industrial revolution*. World Economic Forum, Geneva, Switzerland.