CHAPTER 3

Talking About Algorithms: How Can Interdisciplinary Translation in the Automation of Public Sector Casework Be Facilitated?

Hanne Cecilie Geirbo Oslo Metropolitan University
Rannveig Røste University College Østfold

Abstract: Motivated by the need to save resources, increase efficiency, and reduce human error, public authorities are increasingly developing digital systems for the automation of casework. Since professional practices and algorithmic systems co-evolve, it is crucial that the expertise of caseworkers is included in the design of these systems. This presupposes that the algorithms can be scrutinised and discussed across professional boundaries. Recent literature on the digitalisation of public administration has called attention to several problems of translation associated with the development of algorithms. This chapter discusses two related problems: the problem of transforming laws, and transforming professional practice into algorithms. Based on interviews with system developers and caseworkers in the Norwegian Labour and Welfare Administration (NAV), the chapter presents and discusses tools and methods for overcoming these problems, and facilitating translation between professional groups in the development of digital decision support systems.

Keywords: digitalisation, automation, algorithms, digital decision support systems, public sector casework

Citation: Geirbo, H. C. & Røste, R. (2023). Talking about algorithms: How can interdisciplinary translation in the automation of public sector casework be facilitated? In R. Fugletveit & C. Sørhaug (Eds.), *Lost in Digital Translations: Studies of digital resistance and accommodation to the welfare state in practice* (Chap. 3, pp. 73–90). Cappelen Damm Akademisk. https://doi.org/10.23865/noasp.196.ch3

Introduction

Information and communication technology (ICT) increasingly constitute the frames for how we, as humans, perceive the world and act in it. Despite this computerisation of society, knowledge of how these technologies are constructed and function is mostly reserved for people with expert knowledge of ICT. In the last decade, there have been dramatic advances in the development of digital systems for automating procedures in the public sector, ranging from software that acts on predefined rules, to machine learning where algorithms identify patterns in historical data sets and produce recommendations based on these patterns (Faraj et al., 2018; Janssen et al., 2020; Pencheva et al., 2020; Zuiderwijk et al., 2021). For the public sector, algorithmic systems represent opportunities to improve quality and increase effectiveness in service delivery, but also challenges to the public's trust in government (de Sousa et al., 2019; Zuiderwijk et al., 2021). An algorithm is 'an abstract, formalized description of a computational procedure' that is written into code and applied to data (Dourish, 2016, p. 3). Scrutinising the outcomes of algorithms is difficult, due to the black boxing of input data and rules for processing, or due to the lack of competence in understanding the available information about the algorithms. This is a democratic challenge, because structures of great political and ethical importance may escape public debate (Bowker & Star, 2000; Kitchin, 2017). This is also a challenge to the legitimacy of professional work, since routine tasks are delegated to algorithms, and professional judgement is transformed into decomposed tasks, monitoring, and accountability (Hasselberger, 2019; Orlikowski & Scott, 2014).

As increasing areas of our personal and public lives are being digitalised, a tendency to fetishise algorithms, that is attribute to them powers of their own, has been noted (Ames, 2018; Monahan, 2018; Thomas, 2018). This tendency is expressed and perpetuated in simplified terms and suggestive metaphors, which obstruct informed conversation. Developing vernacular ways of talking about algorithms and other components of digital systems is thus important. Interprofessional teams that develop digital systems for automation are pioneers in this work, since they have to find ways of communicating across areas of expertise, along with these systems being developed and implemented.

In this chapter, we will address the development of digital decision support systems in the public sector and explore how various professional groups talk to each other about algorithms. We approach this interaction as translation between different fields of knowledge and practice, and ask the following research question: How can interprofessional translation be facilitated in order to develop fair, legally sound, and trustworthy algorithms in the public sector?

The chapter focuses on the communication challenges within two related problems of translation. The first problem is the translation from law to algorithms. Developing digital systems for casework in public administration entails making law computational. However, laws are written in a genre that is not directly translatable into the discrete categories required to describe a computational procedure. Adding to this challenge, the translation work is done by programmers who do not have the juridical knowledge needed to assess fully the consequences of their choices. The second problem is the translation of professional practice into algorithms. Caseworkers are requested to delegate tasks to algorithms, which they lack the competence to fully understand.

Given the limited understanding of how various professional groups communicate in developing digital decision support systems, there is a need to study the translation in real life. The empirical basis of this chapter is data from an exploratory case study of how translation problems in the development of automated decision support systems is handled in the Norwegian Labour and Welfare Administration (NAV). NAV is the largest public organisation in Norway and administers benefits to help citizens with labour-related loss of income, such as occupational injuries, sick leave, childbirth and caretaking. NAV has adopted a strategy in which ICT solutions have a central role in channelling and releasing resources to be used in solving complex cases. The selected case is a project to develop a decision support system for a distinct public service in NAV: benefits for the care of sick children. The project has brought together system developers and caseworkers in developing a legal and reliable system. This offers a great opportunity to study in real life how different professional groups talk about algorithms. Further, the chapter will review recent literature on the problems of translation in developing algorithms in the public sector. Then, methodology is discussed, before presenting results from the case study. Finally, the chapter discusses tools and methods for facilitating translation between professional groups in the development of algorithmic systems for casework in the public sector.

Theoretical Approach to Problems of Translation of Algorithms in the Public Sector

Motivated by a wish to save resources, increase efficiency, and reduce human errors, public authorities are increasingly relying on automation in public service systems (de Sousa et al., 2019; Nordrum & Ikdahl, 2022; Pencheva et al., 2020). This entails handling cases by means of processing data from government registers by using computer algorithms. The proportion of automation varies from fully automated systems, in which the entry of data simply produces resolutions, to decision support systems, in which algorithms provide the caseworker with suggested decisions (Scholta, 2019). Systems based on rule-driven algorithms apply predefined if-then codes to settle the outcome of cases, whereas systems based on data-driven learning algorithms can identify patterns in large historical data sets (Bayamlıoğlu & Leenes, 2018; Nordrum & Ikdahl, 2022). Thus, humans are still in the loop in digital decision support systems, but the degree of human involvement varies (Lindgren et al., 2019).

In any case, due to their role in distributing public resources, these automated systems of public sector work constitute infrastructures of great political and ethical consequence (Bovens & Zouridis, 2002). Yet, knowledge of how these systems function is not easily available to the general public, and can be difficult to access and comprehend for the bureaucrats who use them. Understanding how an algorithm has arrived at its outcome can even be unclear to the system developers, due to the quantity and complexity of the input data and its interactions (Faraj et al., 2018; Janssen et al., 2022).

Developing digital systems for automation in public administration entails making law computational, that is formal and quantitatively precise (Hasselberger, 2019; Wihlborg et al., 2016). However, laws are written in a genre aimed at facilitating human interpretation, and are not directly translatable into discrete categories. As described by Kitchin (2017, pp. 16–17), an algorithm consists of two components: the 'logic' component, which specifies what should be done, and the 'control' component, which specifies how it should be done. The logic component is specific to the domain within which the algorithm will work, and requires the translation of a task into pseudocode: a structured formula with a set of rules that establish the categories into which information is sorted. However, system developers rarely have the juridical knowledge needed to assess fully the consequences of their choice of categories. This implies that while the automated systems limit the discretionary power of the caseworkers, such power is allocated to the system designers, who might not be aware of the significance of their power (Bovens & Zouridis, 2002; Lindgren et al., 2019).

The increase in digital automation in the public sector has led to concerns of black-boxing. We can only see the input data and output data to an algorithmic system, and not the process of turning input into output (Ebers, 2022). Black-boxing is particularly problematic in the public sector because its legitimacy is based on how the government executes its tasks in accordance with core values, such as democracy, accountability, and efficiency (Andersson et al., 2018). Automating decisions reduces human bias and increases the likelihood that all citizens are treated equally. However, equal treatment is not always fair treatment. Bovens and Zouridis (2002) ask whether an expert system that leaves no room for considering the specific circumstances of each case can still be considered just. Excessive use of discretion in casework will lead to arbitrariness. However, a system based on the assumption that fair treatment equals uniform treatment can also produce arbitrary outcomes due to excessive rigidity. Hence, to ensure good quality from the decision support systems it is crucial to invite the professional competence of caseworkers into the process of developing them.

The introduction of digital decision support systems implies that the professional practices of caseworkers co-evolve with algorithmic systems (Agarwal, 2018; Grisot et al., 2018). Wihlborg et al. (2016, p. 2903) argue that such systems 'reframe relationships, responsibilities and competences'. They illustrate this through two different strategies that caseworkers can adopt in communication with a client who argues against a decision. The first strategy is to explain why the system arrived at a certain conclusion. The second strategy is to help the client translate information into a format that is better adjusted to the logic of the system, so that the caseworkers and the system can be more precise in arriving at a decision. This illustrates how digital systems for automation do not merely enable or constrain established professional practices, but also engender new professional roles in the interplay with algorithms.

Methodology

This chapter draws on data from an exploratory case study (Yin, 2009). The selected case is a project to develop an automated decision support system for a distinct public service in NAV: benefits for the care of sick children.

An exploratory case study is useful for developing initial understanding through an empirical introduction to a topic of interest. The method follows a theoretical sampling strategy (Eisenhardt & Graebner, 2007), selecting cases to create theoretical constructs of a little-known phenomenon. The case serves as an empirical basis from which to develop theory by the experimental logic of replication: of repeating, testing, and extending the emerging theory in real life contexts. Studying cases in a real-life context is a critical element of case studies, which aim to gather comprehensive empirical material to understand the distinct phenomenon.

The benefit for taking care of sick children is one of several related activities aimed at covering income loss in caretaking situations. The number of applications for this benefit increased enormously during Covid-19, when kindergarten and schools closed, and parents were obligated to stay at home with their children. In general, NAV faced an enormous workload, and the processing time for this and other benefits increased. Before Covid-19, NAV had started several projects relating to digitalisation, and they now considered the benefit for the care of sick children as a suitable service to consider for automation. An automated system would result in an efficient service with reduced processing time, leaving the caseworker with the manual work of checking and controlling the automated decisions. Besides, the benefit for the care of sick children is one of similar related benefits for caregiving situations, in which NAV saw a potential for automating by using the same rule-driven algorithm. This group of related benefits could then provide them with unique experiences on developing digital decision support systems for casework.

In developing the system, NAV invited caseworkers who had experience from working with the benefit. Some caseworkers were released from their daily tasks so they could contribute as experienced consultants in the project group developing the system. All other caseworkers were invited to post questions and comments on their experiences in applying the digital decision support system in their daily work, onto a digital platform. The posted experiences were discussed in project meetings with various professional groups present, such as system developers, project owners, designers, and lawyers. In these discussions, the caseworkers who were enrolled in the project served as translators of their colleagues' casework practice in developing the automated system.

Studying this interprofessional translation work, we applied a narrative strategy, collecting stories of people participating in developing the digital decision support system for the care of sick children. The main source of data collection was interviews with system developers, product managers, and caseworkers. Their narratives were supported, questioned, and put into context through using publicly available information on NAV's strategies and work, their attention to digitalisation, and experienced pressure during Covid-19. Besides, one of the authors had collected data in a previous case study of NAV's effort to develop internal competence on artificial intelligence and digital support systems. This study served as a pilot study for the choice of research design in the study presented in this chapter, offering critical empirical and theoretical insight into the phenomenon.

The primary data material consists of semi-structured interviews with people participating in the project. There were five informants with various professional backgrounds and roles: system developers, product managers, and case workers. We used a number of documents as supporting material: reports from previous projects on digitalisation in NAV, and strategy documents. We also used publicly available information from NAV's own digital news arena MEMU, podcasts, and daily newspapers. Three of the interviews were done between June and September 2022, and two in March 2023. The interviews were based on semi-structured interview guides, focusing on their various roles and tasks in the development project, how they worked with algorithms, and how they talked with people from other professions. In particular, we asked about challenges they experienced in translating their work to people with another professional background, and their tools and methods for overcoming these. The interviews were recorded and transcribed.

The data analysis followed the analytical strategy of replication logic, in which existing theory is used as a template to compare and contrast empirical findings (Eisenhardt & Graebner, 2007; Yin, 2009). We started by selecting interesting statements from the interview material, assessing similar statements, and testing for theoretical patterns. We revised our findings by discussing and sending the analysis back and forth between the authors, and refining our results and final findings. The analysis did not follow a strict deductive style of replication, but iterated between inductive and deductive approaches, where data collection was inspired by previous data. In interpreting data from the interviews, we arrived at new insights about related concepts, which we decided to investigate further, and which led us deeper into the material. These concepts were related to how various professional groups talk about algorithms in digitalising public sector case work, and the related problematic issues of translating of laws and professional practice, which is addressed in the scholarly literature. We gained critical insight into the conceptual aspects of interprofessional translation, through the various professional groups involved in the project. Also, the technical aspect of digital decision support systems was investigated, which in this project turned out to be strictly rule-based algorithms and not data-driven learning algorithms, which we as researchers thought it would be. This empirical insight into digitalisation technology led us to an extended review of the concept of algorithms in social science literature.

Translating Law into Algorithms

In the interviews, the system developers and the product managers describe the development of information systems for decision support as consisting of many concrete operations of programming. The procedures construct so-called stopping points on each formal requirement in the legal basis for the public service:

> The team consider all the relevant laws and ensure that everything is in order, for example has the applicant applied within the deadline? Does the applicant have the right age? Has he lived long enough in Norway? Does he nurse someone? Is there any information from a doctor? Is he an employee, freelance or self-employed? Based on all the information a calculation of the compensation is made. (Product Manager, NAV)

However, as pointed out in the scholarly literature (Hasselberger, 2019; Wihlborg et al. 2016), the problem is that laws are written in a genre not directly translatable into discrete categories. The law is not written for computer programs, as noted by a system developer in one of our interviews:

> The National Insurance Act is poor craftsmanship if you write it as code, because you break some principles by referring to things across chapters. Chapter 9 points very much to chapter 8, which is sick leave benefits. If one is going to refer across, it should be taken out of sick leave benefits and be a separate chapter. (System Developer, NAV)

The system developer points to the challenge of how the information in the National Insurance Act is structured. While a human being who reads chapter 9 can easily follow an instruction to look up a section in chapter 8, such cross-referencing is not easily translated into algorithms that serve as instructions for a computer.

While the translation of laws to algorithms can be a critical challenge, it can also shed light on gaps and inconsistencies that had previously escaped systematic attention. Moreover, our case study showed that in translating laws to algorithms, the system developers became aware of new juridical aspects of case management work:

> There are some laws that eliminate each other, and you will first be aware of this when you put the rules into the system. Then you notice that the rule is not possible to implement because the two laws eliminate each other. (System Developer, NAV)

In translating the National Insurance Act into pseudocode one, for example, found that some groups of users had been uncategorised in the previous system:

> One has the right to adjustments for work, for example, 'I will reduce my position by 50%'. But what about the ones who have not had any job, how are they to be categorised? How can you assess a loss of 20% of income for them? How can you assess loss of work when you have not had any work? (System Developer, NAV)

In cases like this, important juridical conundrums requiring clarification are discovered when attempts to describe a task as a structured formula with a set of rules fails. Similar to infrastructural inversion (Bowker, 1994), where action is taken to bring the otherwise transparent or slippery infrastructure into view, the translation from law to algorithms can render inconsistencies in the law 'visible through programming'.

When there are juridical inconsistencies, programmers may end up in a position where they need to prioritise to make the system work. This means that the discretionary judgment previously held by caseworkers may be transferred to system developers (Bayamlıoğlu & Leenes, 2018; Bovens & Zouridis, 2002; Lindgren et al., 2019). This redistribution of discretionary power can result in important decisions being taken unknowingly and without auditable traces.

The detailed step by step operation of programming in the development of new case management systems in NAV has resulted in many discussions of the laws, interpretations, and inconsistences. This translation work is done by system developers who do not have juridical training. However, in translating laws into algorithms they interact with other groups of people who do have specified knowledge about the legal basis of public services, such as the product managers, the Ministry of Labour and Social Inclusion, and the caseworkers, etc. Our data material shows that interaction with caseworkers is crucial in translating algorithms, but also difficult.

Translating Professional Practices into Algorithms

Digital decision support systems imply automation of casework, aimed at standardising simple cases and releasing resources to attend to complex cases (Larsson & Haldar, 2021; Scholta et al., 2019). This may sound logical and uncomplicated, but in practice it involves many possible transition failures (Bayamlıoğlu & Leenes, 2018; Nordrum & Ikdahl, 2022). Translation between different logics of problem solving is one of the challenges. The new system introduces a step-by-step procedure, in which the caseworker is guided through information collected from various public records, like the population register, income and tax information, medical diagnosis, etc.

> The system collects the necessary information needed for the case management and presents the relevant information for each decision to the caseworker. (System Developer, NAV)

Our data material indicates that this step-by-step approach represents a radical break with the previous practice of many caseworkers:

> Many of the proceedings in the past have been in people's heads: that you read an application and then make up your mind, and then you grant benefits according to that. But (the new system) splits up the casework, you could say. Based on the information it collects, you can stop at various action points. (Caseworker, NAV)

A holistic approach to case handling, in which the caseworker establishes an overview of the case before delving into the details, cannot be practiced with the new system. While there is still room for using discretion in the new system, it is not the case worker but the system that decides when discretion can be used. Some experience this as a loss:

> They lose control when the system handles the process. They feel that they do not own the case anymore, because they are just asked to do small tasks. 'Control the letter', and such things. They have lost everything they felt was casework. (System Developer, NAV)

However, some people might be inclined towards an algorithmic approach to problem solving, whereas others might be more intuitive and holistic in their casework. For the latter group, it will be harder to adjust to an algorithmic system.

The implementation of digital decision support systems is not only changing how caseworkers understand their own professional role, but is also, as Wilhborg et al. (2016) put it, reframing relationships, responsibilities, and competences that the caseworkers have in relation to others. When a new technology is introduced in a workplace, this can alter the established hierarchies and change power dynamics (Faraj et al., 2018). With increasing automation, advice from a newly employed colleague who masters the technology might be more in demand than the experience-based knowledge of long-term employees. This was expressed in one of the interviews:

> … those who seem to find this the most demanding are perhaps those who have previously been very good at their profession, and had been the one everyone asked. Now they are suddenly in a completely different situation where they may have to ask the newer, or younger colleagues. The roles are, in a sense, completely reversed. (Caseworker, NAV)

The introduction of automation also accentuates the relationship between the organisational units. Some caseworkers interpret the delegation of their tasks to algorithms as a signal that their work is no longer trusted:

> [Some] experience these changes as meaning that they had done everything wrong before. 'Why can't we do it like this, don't you trust us? Don't you trust that we can manage this?' (Caseworker, NAV).

However, it is of critical importance to involve caseworkers in the development of the algorithms. Implementing new information systems in an organisation entails grappling with existing practices and conventions that can inhibit change but also be a key to successful adoption if used as a resource in the development process (Aanestad et al., 2017; Star & Ruhleder, 1996). Blurring the line between the development phase and the use phase has its risks, because the system that is released for use will necessarily contain errors. The timing of when to release a new module of the system is important, but tricky. If you release a module too late you lose important testing opportunities, but if you release it too early the amount of error can erode the trust that caseworkers have in the system:

> Trust is so easy to say but so hard to earn. If you've done something that causes you to lose it, it takes a long time to get it back. It is a bit of a challenge to put new systems in motion, because new systems often have errors, and when something is wrong, trust falls. You will not be able to create anything flawless from day one. (System Developer, NAV)

Involving caseworkers is not only essential for assuring the quality of the algorithms, it is also crucial for developing the caseworkers' understanding of how algorithms work. As emphasised in a report from the Norwegian Data Protection Authority (2022), insight into and understanding of how the algorithms work is important for the caseworkers' ability to assess critically the recommendations they produce. While building trust in the system is crucial, it is also important to prevent 'automation bias', the blind belief that the computer is always right (Carr, 2014; Hasselberger, 2019).

NAV has established several arenas for involving caseworkers in developing the automated systems. There are digital communication channels where caseworkers at the NAV offices can ask questions and seek guidance when they encounter problems. Since these channels facilitate dialogue, they allow opportunities to tailor explanations to the needs of individual caseworkers. In addition to helping build knowledge about the systems among the caseworkers, these channels are also important for detecting gaps and errors in the solutions. Caseworkers also interact with system developers in the development project. Our case study shows that communication about algorithms between caseworkers and system developers is challenging, but that NAV uses several tools and methods to facilitate translation between these two groups. In the next section we will discuss some of them.

Tools and Methods to Facilitate Interprofessional Translation in System Development

Interaction between caseworkers and system developers is essential for developing well-functioning algorithms for digital decision support systems, but since they contribute to this work with different knowledge, their contributions are likely to be characterised by partial understanding. One of our interviewees emphasised that distinguishing between what is necessary and not necessary to understand is important for effective communication across various professional groups:

> New people in the team often have problems understanding how the developers talk. They talk about things like Java and Jakarta, and you don't understand what they are. But now that I have worked with the developers for a long time, I no longer think about the things that I don't understand. Now I distinguish between what I need to understand and what I don't need to understand. (Product Manager, NAV)

This also applies to caseworkers at local offices:

> Think about a telephone for example. You can use it without needing do know what is inside it. As a caseworker you have to understand the Proceedings Act, but you do not need to know that Kafka is used for developing the system. (Product Manager, NAV)

While striving to understand professional secrecy can be counterproductive, having an overall understanding of the perspectives and concerns of the different professions is important for working together and collaborating on development projects. The following is a reflection of a data scientist on his collaboration with lawyers and designers:

> While we are not lawyers, we need to have a sufficient understanding of law, of what you want to safeguard, what you mean by this question, what motivates this question. Because when a question comes from a designer, and when it comes from a lawyer, there are often two different things they want to safeguard. Both want to create good services, but the starting point is different. (Data Scientist, NAV)

The system developers seek to bridge the professional communication gap by using terms and concepts that are familiar to the caseworkers:

> We actually talk about it in the same way as what you see. There is always a cut-off point, because this is jargon that the caseworker recognises. The cut-off point is when you are first entitled to the benefit. So we use the same jargon as the caseworker. The calculation basis, and things like that. There is, in a way, a catalogue of terms that exists. It makes sense to reuse [the terms], because we then have a clear language. (System Developer, NAV)

Communication is facilitated by framing the unknown in known terms. Thus by reusing the terms the interprofessional group can build a shared vocabulary over time. Talking about the algorithms in terms that are specific to what NAV does not only benefits the caseworkers' understanding, but also serves the purpose of maintaining a common focus on the organisation's overall aim. NAV is an attractive workplace for system developers, because of the opportunities to develop advanced technical solutions, but interest in technical issues should not overshadow the purpose of developing the systems:

> NAV is supposed to have interdisciplinary teams that will solve the user's needs. Everyone is expected to do so. It is important to be aware that the purpose of creating solutions is not the technical, but the functional. I expect the developers to be able to talk functionally about things. (Product Manager, NAV)

Another tool used for facilitating translation between developers and caseworkers is visualisation:

> I'm a fan of drawing, trying to visualise where the problem is, and how it will turn out for the different groups. So a visual and good dialogue is essential. (System Developer, NAV)

Since algorithms are logically structured instructions with entry points for input, application of rules, and production of output, they lend themselves easily to visualisation:

> … when we try to visualise for professionals what the flow is like through the system, and how specific rule types are to be implemented, it is usually decision trees or things like that, which can clearly depict the flow. Where does someone fall out in a rejection, which criteria go into a rejection? (System Developer, NAV)

Much interprofessional translation happens before the system is released for use, but some needs for translation also emerge when the algorithms become part of the everyday practice of the caseworkers. One example concerns errors in the input to the algorithms, such as clients' applications. Initially, there was no opportunity to correct erratic input, but soon after caseworkers started to use the system, the need for incorporating this practice emerged. This resulted in a support system named Punch:

> So if something is wrong and we want to correct something, we have Punch, and then we can punch in the information we receive so that it overrides the system. It wasn't there at the start, but it is absolutely necessary, because it happens all the time that clients make mistakes when filling in forms. (Caseworker, NAV)

This highlights time as an important dimension of interprofessional translation. Mundane, but essential practices can easily escape the attention of a developing team and will first emerge after the system is in use.

Concluding Remarks

Motivated by the wish to increase efficiency, save resources, and reduce human errors, systems for automating casework are increasingly used by public authorities. Despite the important role such systems have in distributing public resources, knowledge of how they are constructed and function is difficult to access for the general public, as well as for the professionals who are asked to rely on them in their casework. Moreover, those who develop these systems often lack the competence to assess fully the consequences their programming will have for casework. Based on a case study of the Norwegian Labour and Welfare Administration (NAV), we have discussed problems of translation from law and professional practice into algorithms, and explored tools and methods for facilitating interprofessional translation in the development of automated decision support. To conclude this chapter, we will suggest some recommendations based on our findings for how interprofessional translation in the development of automated systems can be facilitated.

Establish low-threshold communication channels. Involving caseworkers in system development is essential for quality assurance and error detection, and also for developing their ability to assess critically the recommendations produced by the algorithms, so automation bias can be avoided. Frequent contact with a wide range of caseworkers can be facilitated through low threshold communication channels on digital platforms.

Distinguish between what needs translation and what does not. While some common ground is necessary for translating between professional practice and algorithms, it is also important to identify what one does not need to understand. Competence in coding is, for instance, not necessary for caseworkers to be able to contribute their professional expertise in developing pseudocode.

Use domain-specific language and visualisation. Using vocabulary from casework to talk about algorithms is not only useful to develop automated systems, but can also strengthen the system developers' commitment to the functional purpose of the system. Decision trees and other visual aids are useful for showing and discussing how algorithms work.

Allow time for translation needs to emerge. Some translation needs will emerge through practice. Programming can render inconsistencies in the law visible, and errors and needs for alterations will be revealed when the system is applied in casework. Hence, it is important to set aside time and resources to make the necessary adjustments after the system is released for use.

With the rapid digitalisation of increasing areas of public and personal life, 'algorithms' has become a catchword in public debates, referring to a vaguely defined set of processes that concern the delegation of tasks to digital technology (Thomas et al., 2018). As noted by several scholars (Ames, 2018; Monahan, 2018; Thomas et al., 2018), there is a risk of fetishising algorithms, in the sense of attributing to them power of their own and treating them as 'magic black boxes' (Thomas et al., 2018). This can lead to knee-jerk rejection of any algorithmic system, but also to deterministic responses, in which technological development is seen as inevitable, and critical debate therefore seems futile. To cultivate a broad, informed debate on digitialisation in the public sector, there is a need to facilitate vernacular conversations about the inner workings of digital technology, such as algorithms. Translation practices in organisations at the forefront of developing digital public service systems could inspire approaches to initiating inclusive and constructive dialogue on algorithmic systems in other areas of society as well.

References

Aanestad, M., Grisot, M., Hanseth, O., & Vassilakopoulou, P. (2017). Information infrastructures and the challenge of the installed base. In *Information infrastructures within European health care* (pp. 25–33). Health Informatics.

Agarwal, P. K. (2018). Public administration challenges in the world of AI and bots. *Public Administration Review*, *78*(6), 917–921. https://onlinelibrary.wiley.com/doi/full/10.1111/puar.12979

Ames, M. G. (2018). Deconstructing the algorithmic sublime. *Big Data & Society 5*(1), 2053951718779194. Sage Publications.

Andersson, A., Hedström, K., & Wihlborg, E. (2018). *Automated decision-making and legitimacy in public administration*. http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-73989

Bayamlıoğlu, E., & Leenes, R. (2018). The 'rule of law'implications of data-driven decision-making: A techno-regulatory perspective. *Law, Innovation and Technology*, *10*(2), 295–313.

Bovens, M., & Zouridis, S. (2002). From street-level to system-level bureaucracies: How information and communication technology is transforming administrative discretion and constitutional control. *Public Administration Review*, *62*(2), 174–184. https://doi.org/10.1111/0033-3352.00168

Bowker, G. C. (1994). *Science on the run: Information management and industrial geophysics at Schlumberger: 1920–1940*. The MIT Press.

Bowker, G. C., & Star, S. L. (2000). *Sorting things out: Classification and its consequences*. The MIT Press.

Carr, N. G. (2015). *The glass cage: Where automation is taking us*. The Bodley Head.

Datatilsynet (The Norwegian Data Protection Authority). (2022). *Sluttrapport fra sandkasseprosjektet med NAV*. https://www.datatilsynet.no/regelverk-og-verktoy/sandkasse-for-kunstig-intelligens/ferdige-prosjekter-og-rapporter/nav-sluttrapport/

de Sousa, W. G., de Melo, E. R. P., Bermejo, P. H. D. S., Farias, R. A. S., & Gomes, A. O. (2019). How and where is artificial intelligence in the public sector going? A literature review and research agenda. *Government Information Quarterly*, *36*(4), 101392. https://www.sciencedirect.com/science/article/pii/S0740624X18303113

Dourish, P. (2016). Algorithms and their others: Algorithmic culture in context. *Big Data & Society*, *3*(2), 2053951716665128.

Ebers, M. (2022). Regulating explainable AI in the European Union: An overview of the current legal framework(s). In L. Colonna & S. Greenstein (Eds.), *Nordic yearbook of law and informatics. 2020–2021: Law in the era of artificial intelligence.* The Swedish Law and Informatics Research Institute. https://irilaw.org/2022/02/16/new-publication-nordic-yearbook-of-law-and-informatics-2020-2021/

Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, *50*(1), 25–32. https://journals.aom.org/doi/abs/10.5465/AMJ.2007.24160888

Faraj, S., Pachidi, S., & Sayegh, K. (2018). Working and organizing in the age of the learning algorithm. *Information and Organization*, *28*(1), 62–70. https://www.sciencedirect.com/science/article/pii/S1471772718300277

Grisot, M., Parmiggiani, E., & Geirbo, H. C. (2018). Infrastructuring internet of things for public governance. *Research-in-Progress Papers*, *66*. https://aisel.aisnet.org/ecis2018_rip/66

Hasselberger, W. (2019). Ethics beyond computation: Why we can't (and shouldn't) replace human moral judgment with algorithms. *Social Research*, *86*(4), 977–999.

Janssen, M., Hartog, M., Matheus, R., Yi Ding, A., & Kuk, G. (2022). Will algorithms blind people? The effect of explainable AI and decision-makers' experience on AI-supported decision-making in government. *Social Science Computer Review*, *40*(2), 478–493.

Kitchin, R. (2016). Thinking critically about and researching algorithms. *Information, Communication & Society*, 1–16.

Larsson, K. K., & Haldar, M. (2021). Can computers automate welfare? *Journal of Extreme Anthropology*, *5*(1). https://doi.org/10.5617/jea.8231

Lindgren, I., Madsen, C. Ø., Hofmann, S., & Melin, U. (2019). Close encounters of the digital kind: A research agenda for the digitalization of public services. *Government Information Quarterly*, *36*(3), 427–436.

Monahan, T. (2018). Algorithmic fetishism. *Surveillance & Society*, *16*(1), 1–5.

Nordrum, J. C. F., & Ikdahl, I. (2022). En vidunderlig ny velferdsstat? Rettsstaten møter den digitale velferdsforvaltningen. *Tidsskrift for Velferdsforskning*, *25*(3), 1–19. https://doi.org/10.18261/tfv.25.3.1

Orlikowski, W. J., & Scott, S. V. (2014). What happens when evaluation goes online? Exploring apparatuses of valuation in the travel sector. *Organization Science*, *25*(3), 868–891. https://pubsonline.informs.org/doi/abs/10.1287/orsc.2013.0877

Pencheva, I., Esteve, M., & Mikhaylov, S. J. (2020). Big data and AI: A transformational shift for government. So, what next for research? *Public Policy and Administration*, *35*(1), 24–44. https://journals.sagepub.com/doi/full/10.1177/0952076718780537

Scholta, H., Mertens, W., Kowalkiewicz, M., & Becker, J. (2019). From one-stop shop to no-stop shop: An e-government stage model. *Government Information Quarterly 36*(1), 11–26. https://doi.org/10.1016/j.giq.2018.11.010.

Star, S. L., & Ruhleder, K. (1996). Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research*, *7*(1), 111–134.

Thomas, S. L., Nafus, D., & Sherman, J. (2018). Algorithms as fetish: Faith and possibility in algorithmic work. *Big Data & Society*, *5*(1), 2053951717751552.

Wihlborg, E., Larsson, H., & Hedström, K. (2016). *"The computer says no!" A case study on automated decision-making in public authorities* (pp. 2903–2912) [Conference Presentation]. 2016 49th Hawaii International Conference on System Sciences (HICSS).

Yin, R. K. (2009). *Case study research: Design and methods* (4th ed., Vol. 5). Sage.

Zuiderwijk, A., Chen, Y. C., & Salem, F. (2021). Implications of the use of artificial intelligence in public governance: A systematic literature review and a research agenda. *Government Information Quarterly*, *38*(3), 101577 https://www.sciencedirect.com/science/article/pii/S0740624X21000137