

Law and algorithms in the public domain

Dag Wiese Schartum

Norwegian Research Center for Computers and Law, Faculty of Law, University of Oslo
d.w.schartum@jus.uio.no

DOI: <http://dx.doi.org/10.5324/eip.v10i1.1973>



This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

This article explains and discusses the relationship between traditional legislative processes and the development of automated government decision-making systems. It could be argued that certain juridical aspects of the systems development process should be regarded as quasi-legislation. The author investigates and discusses possible ways of changing this process with a view to increasing and improving openness and political involvement in tasks today often regarded as mere implementation.

Keywords: Algorithmic law, automation, government administration, open domain, legislation

Introduction

Most laws are manually implemented and based on the presumption that the text will be read, understood and followed on a case-by-case basis. In some parts of legislation, particularly within various branches of government, the expectation is that legislation will be implemented by means of automated decision systems. If so, the first thing that will happen after enactment is that a group of experts will read and interpret the law “once and for all” and transform it into a computer system designed to automate individual decisions. Such government systems will imply that processing of individual cases will be processed according to predefined algorithms, and with limited human involvement, if any. In order to establish such systems, the relevant government agency must initiate a systems development process.¹ An important part of this process is embedded legal decision-making that may be seen as “hidden” quasi-legislation, representing processes and decisions that are only recognised and graspable by the few initiated. In this article, I will describe and discuss regulatory aspects of administrative laws that are digitalised and automated. More specifically, I will discuss the relationship between the traditional legislative process on the one hand, and the development of information systems to implement legislation on the other.² The discussion concerns relationships between open political processes according to the legislative process, and the emergence of closed legal processes which form part of systems development. In concluding discussions, I discuss possibilities of changing the legislative process in order to foster political involvement and openness in tomorrow’s government administration.

Digitalisation of legislation and the development of legal decision-making systems

Important parts of administrative legislation are drafted with the knowledge and expectation that post-enactment implementation will largely rely on digital systems as a means of achieving full or partly automated application. In such systems, algorithms containing a mix of prescribed arithmetical and logical operations will control every automated decision process.³ At this stage of technological development, most digitalised legislation is found within mass administration, typically relating to social welfare schemes, pensions, taxes and excises, etc.; i.e. legislation dealing with the distribution and redistribution of wealth in society. Mass administration denotes government administration established to process large numbers of individual cases, typically by means of highly automated processing. The legislation fed into such mass administrative systems is often highly complex,⁴ and the combination of very high numbers of individual cases and legal complexity places great demands on the administrative machinery to deliver swift processing, equal treatment, a large degree of predictability, etc. Government agencies, typically directorates, develop and maintain these systems – legal decision-making systems – that are designed to process individual cases and conclude them with legally valid decisions without (major) involvement by human officers.

The existence of legal decision-making systems implies that the relevant government agency collect all relevant legal sources, interpret and deduce acceptable legal rules from this material, and express these rules by means of programming language.⁵ I denote the process in which legal sources expressed in natural language are reshaped into formal representation of the law by means of programming language as *transformation of legal sources* (or “transformation” for short).⁶ In Norway, such transformation is usually carried out by a project team composed of representatives from the government agency in charge of the system and external experts.⁷

Transformation is only one part of the complex process of systems development. The development process contains a series of different and largely interrelated questions, in which establishing legal rules and expressing them in programming language is one.⁸ Here, I will limit the discussion to the process of transformation of legal sources. As a result, transformation must have clear instructions to the computer, meaning that the law expressed in natural language with all its vagueness and ambiguities must be transformed into precise sets of instructions. This implies a shift from natural language to programming language. Since programming languages are precise and thus unambiguous, transformation implies that uncertainties and flexibility in the interpretation of legislation are replaced by accurate and fixed machine-readable sets of legal rules.

In the course of a transformation process, several legal choices are made which may have significant consequences for the processing of individual cases and the substantive outcome of each case. Margot (1991: 20) underlines that “It is necessary ...to anticipate all possible answers to all questions, and all possible interactions among these answers ...”. Crucial questions lie in the understanding of the concepts by which the law is formulated, and in particular the concepts that denote facts of individual cases. Many such concepts are not defined or are only partly explained.⁹ For instance, the text of the law uses concepts like “annual income”, “student”,

“employee”, “cohabitant”, “retired person”, etc. After enactment, in the course of the systems development processes, many such concepts may be rigidly defined. The chances of this happening are particularly high if one of the administrative aims is to base processing on automatic collection of data. Provided available data definitions are within the margin of interpretation, the result will easily be that available data from machine-readable sources are preferred. The effects of such choices could be considerable. “Retired person” for instance, could be interpreted as indicating membership of public and private pension schemes. These could include large general pension schemes and a variety of small, specialised schemes, and the same person could be a member of two schemes or more. As part of the development of an automatic system for the processing of cases concerning retired persons, it will be up to the government agency in charge to select a combination of data sources from existing pension schemes that could be said to correspond to the concept of “retired person” in the relevant implemented law. Choices will be taken within the existing margin of interpretation, and in light of possible access to machine-readable data sources.

Similarly, as part of transformation, possible uncertainties concerning the logical structure of the law will need to be clarified: In which sequence should conditions of the law be carried out; are conditions alternative or cumulative; how should the described computations be understood, etc. Moreover, legislation may be silent on certain legally relevant problems; decision-making systems on the other hand must not contain any “blank spots”. Thus, development of decision-making systems often requires that the law is supplemented. For instance, as part of describing a benefit or obligation, law uses the concept “week”, but is silent regarding the number of days in a week, i.e. if the correct understanding is a seven-day calendar week or a six- or five-day workweek. Such detailed questions could obviously have great effects on individual decisions, for instance as part of the calculation of benefits and taxes.

Complete automation means that both the collection of data describing case-relevant facts and the processing of these data are automated. In this case, the system is programmed to access relevant case-related facts from predefined machine-readable sources. Thus, it will for instance automatically access basic biographical data from the Population Register, income figures from the tax administration, data regarding pensions and social benefits schemes from national labour and social services, health information from medical services (identified by means of doctor and patient registers), etc. All collected data will then be processed according to algorithms established on the basis of the legal rules describing how the facts of cases result in individual decisions. However, as Margot (1991: 21) underlines: “Legal provisions are seldom expressed as detailed algorithms, and the task of converting even a small fragment of legislation into this form can be so daunting that it becomes practically impossible.” Thus, mass administrative decision systems are often results of such super-complex interpretation and representation processes of relevant legislation, and it is this algorithmic representation – and not the authentic law – which is applied to individual cases and which decides results.

In brief, transformation from authentic legal texts to rules expressed in computer programs is about trying to understand law as a question of executing logical and arithmetical operations on certain well-defined data. Trying of conditions (i.e. execution of logical operations) and computations¹⁰ (i.e. execution of arithmetical

operations) represent two basic types of processing rules, and rules that cannot be represented in this way, e.g. discretionary rules, may not be represented in the program as discretions.¹¹ The computer program will, for instance, try cumulative and alternative selection criteria for admission to a university study programme, and as part of this process calculate points according to reported grades from relevant schools, applicant age, etc. Full automation implies that decisions will be made only on the basis of predefined, machine-readable case-relevant information, processed according to established algorithms.

In addition to programming code determining substantive contents of individual decisions, transformation and formal representation of the law is pivotal for generally relevant questions such as data and privacy protection, information security, degree of reuse of collected data, how processing should be organised, level of automation, role of citizens (including self-service), etc. Thus, systems development and transformation of law into decision systems in government administration are not only about very detailed establishment of existing procedural and substantive rules; they also address various other relevant questions about the design of tomorrow's government administration – from core to surface.

Although we may view transformation processes as interpretation and application of the law, transformation deviates strongly from traditional application in at least two ways: Firstly, transformation requires a more or less complete determination and representation of legal rules that could be deduced from provisions of the Act, relevant case law, etc. When lawyers apply legislation on a traditional case-by-case basis, they need not identify all potential questions of interpretation. In fact, some potential interpretation issues may never be revealed before Acts are repealed. In contrast, transformation and development of mass administrative decision systems require systematic mapping and resolution of “all”¹² possible interpretation alternatives; the system must be capable of processing any individual case in the relevant legal area. Secondly, when we apply legislation traditionally, case by case, interpretation occurs over a long period – usually years and even decades – and many questions are not resolved before they actually occur in an individual case. With a system-driven approach, we need to resolve every question of interpretation before the system is put into use and prior to experiences from real cases.

Transformation of law into computer programmes with detailed statements in the code determining each and every question of interpretation could be seen as resembling traditional strategies of issuing circular letters with instructions and guidelines on how the law should be applied. However, there are at least two significant differences: Firstly, while circular letters typically will be based on experienced interpretation problems etc., legal instructions in programming code will be established before implementation and thus prior to such experiences. Secondly, while circular letters may be read, followed or ignored by human officers in charge of individual cases, automatic processing will always be in compliance with the legal rules embedded in programming code; machines do not disobey or forget.¹³

Even though these transformation processes formally are about the application of the law, they may alternatively and perhaps with greater justification be regarded as resembling legislative processes: Transformation is about establishing precise rules for the digital processing of individual cases. If the degree of automation is high, it

predestines future decisions to varying degrees in all individual cases covered by the system. In Norway, the first example of a fully automated legal decision-making process in government administration – with no elements of human assessment – dates back to the Housing Benefit System of 1972.¹⁴ Today, a number of central decision-making processes are highly automated without any manual interference in “trivial” cases.¹⁵ As a result, for the great majority of cases, processing devoid of human assessment is a reality. Surveys of expectations regarding automated decision-making in Norwegian central government administration show that government agencies expect the degree of automation to rise.¹⁶ Degree of automation vary greatly both among and within countries. Nonetheless, it is likely that many Western-European countries will continue on a path towards greater automation of processes and decisions established by law. The current situation and likely developments in countries that already have a high degree of automation call for a discussion of the relation between legislation and implementation by means of automated decision-making systems.

Black boxes

Legislative processes are fundamentally open in the sense that the process is known, and that principal proposals, arguments and conclusions are publicly available. In contrast, unless accessible documentation of legal content is produced, interpretation decisions regarding transformation as described above are made behind closed doors and “canned” in technical systems.¹⁷ If so, outsiders cannot observe the legal contents and are excluded from easy access to exact knowledge of legal contents. When only process inputs and outputs are observable and the internal structure or function is not understood, we may employ an extended black box metaphor. Within computing, “black box” may describe a situation where we are only able to observe inputs and outputs; what really happens to the inputs is in the dark.¹⁸ In completely automated decision-making systems, every input and most outputs are encapsulated in the box too: The information sources representing input are not observable. We can only see particular results, but unless we are in a position to “open the can” we cannot know all the types of outputs. Individual parties may for instance observe decisions in their own cases, but cannot necessarily access other outcomes, such as case information that is stored in databases available to certain sets of people, that the system shares copies with external authorities and businesses, etc.

In democracies under the rule of law, it is basic that people should enjoy openness and the possibility to make up their own mind and take issue if they disagree with exercise of government powers. Thus, regardless of whether we see the described transformation processes as application and implementation of the law or as quasi-legislation, access to information regarding the detailed rules in the system should be publicly available. How can we avoid “black boxes”?

Outcomes of the transformation process, i.e. the general rules that will be applied in each individual case to be processed by the system, may be expressed in at least three ways: Firstly, rules laid down in the programming code (regarding data definitions, rules controlling exact processing of data, etc.) may be expressed in natural language (e.g. English). In practice, this would for instance imply that the government agency makes a selection of the most important legal choices made in

the course of systems development. An obvious advantage of such a line of action is that described rules are generally intelligible and may easily be made publicly accessible. The downside of such a technique is that statements in natural language are less precise than the programming statements that they are based on. Secondly, rules embedded in computer programmes may be expressed by means of requirement specifications describing legal contents in a semi-formal manner (pseudocode).¹⁹ Such specifications are primarily bases of programming, but when it is confirmed that programming is completed in compliance with specifications, pseudocode expresses actual applicable rules. An important advantage of using pseudocode to express rules of the automatic system is the accuracy and close connection to the rules actually implemented. A disadvantage is that pseudocode statements are much more detailed and comprehensive than e.g. natural language statements as mentioned above. Thus, pseudocode is primarily suitable for experts and people with specific needs to understand legal peculiarities of the system. Thirdly, the programming code will express the legal contents of the system in a complete and 100% accurate way, and display of the code will give precise and full information to everybody who could read it. However, obviously, programming code is not written to be read by people,²⁰ and will only have value for a very small group of programming experts. Hundreds or thousands of lines of programming code are not likely to create much openness.

Changed legislative model?

A gap between traditional legislation and computerised implementation arises when the legislator fails to realise the needs of modern digital governance and, even though it is clear that implementation will be highly automated, formulates rules as if there would in fact be human officers in charge individually considering each case, interpreting wording of the law, performing discretionary assessments, etc. When legislators expect and support automated decision-making they should to a large extent also consider important legal substantial effects of this way of exercising government powers. In other words, legislation should fit the actual needs of automated decision processes.²¹ “Computer-conscious law-making” was suggested as early as the 1970s, but has never been high up on the agenda.²² Compared to 40 years ago, legislation is of course considerably more computerised today, and implications of the gap between legislation and computerised implementation are much greater today than before. Thus, in this author’s view, we need to reconsider the process from legislative drafting to programmed law. Here, I will not discuss detailed techniques of computer-conscious law-making, but will only discuss three overall models that could give room for such an approach.²³

The first possibility is to upgrade the transformation process without adapting the preceding legislative process. Upgrading can involve giving formal status to legal interpretations forming the basis of programming, and designing procedures to safeguard a high level of openness and democratic involvement in the transformation process. Governments could for instance instruct the relevant government agency to establish such fixed rules as they find reasonable and requisite for fair and effective implementation. Moreover, government agencies could be obligated to document these rules and make them publicly available. In case, proposed detailed rules should be open to legal review by courts of law.²⁴ In

other words, it could be made clear that even alternative interpretations to those implemented in the system may be seen as representing legally defensible understandings of the law. The drawback of such an arrangement is the danger of having a high number of complaints, and thus reducing the efficiency effects of automated decision-making. If only a small percentage of taxpayers or social insurance benefit recipients lodged complaints asserting that a government system is based on incorrect or inadequate understanding of the law, this would in itself imply a huge and expensive administrative burden. Thus, I assume that in many cases such a situation will be undesirable and even impossible to cope with.

A *second possibility* is to transform major parts of what is currently viewed as implementation into procedures of delegating and implementing legislation. By this, I mean binding, secondary legislation that supplements the substantive and procedural contents of primary legislation. Documentation and accessibility of non-binding rules of implementation, as described above, only represent disputable interpretations of the law that could be deviated from in individual cases. In contrast, delegating and implementing legislation imply that we see legal rules embedded in decision-making systems as the exercise of general, binding government powers.

Delegating and implementing powers should be based on clear legal authority that establishes and limits the government agencies' authority to make legally binding choices for effective and reasonable digital implementation of the law. Moreover, such legal authority should establish the necessary procedural requirements to guarantee sufficient openness and possibility of legality control of the exercise of authority. Examples include passing rules on how data sources may be used; including data definitions; updating requirements, etc.; and any new rules required to cover blank spots in the primary legislation.

Power to pass secondary legislation on definitions etc. could possibly form the basis of a public review of bills where citizens, alternatively a limited list of stakeholders, are given the right to object and suggest alternative solutions. If, for instance, it is proposed to automatically collect and apply domicile data from the Population Register, but stakeholders assert that data updating routines are insufficient for the required use, stakeholders should have the power to demand use of other or supplementary data sources. In the case of such complaints over proposed implementation strategies, the final decision could be taken at the appropriate political level, for instance in the relevant Ministry.

A *third alternative* would be to change the primary legislative process itself, by incorporating resolution of technological and organisational questions into the normal legislative process. Such a strategy could imply that most of the necessary legal, technological and organisational solutions take place prior to enactment. With this legislative strategy, expert consideration of regulatory questions would basically be carried out as per usual, but with an additional element of formal analysis of proposals in the draft. If the legislature suggests rules such as the one cited below, an analysis team could investigate associated possibilities and challenges. For instance, it is proposed to define "partner" as

a spouse, civil partner or one of a couple whether of the same sex or opposite sex who although not married to each other are living together and treat each other as spouses. (The Well-being of Future Generations (Wales) Act 2015).

The team tasked with formal analysis would investigate the technological-administrative aspects of such proposals. In the example above, one such issue would be examining whether a machine-readable source exists which is based on the right definition of “partner”, and with accurate, reliable and sufficiently upgraded data. If not, the next question might be if a machine-readable database exists with an alternate but acceptable definition of the concept. In the event of a negative answer to this second question, a possible conclusion is that a new register would have to be established as part of implementation. Given the obvious administrative effects this would have, the formal analysis team would leave it up to the experts to decide whether or not the originally proposed definition should be accepted. It would appear that the cited definition presupposes scrutiny of each relationship between possible partners: How could it be established that they really “are living together” and “treat each other as spouses”? What will the administrative consequences and costs be if the definition is enacted? The team may assess these consequences and suggest possible alternative definitions which may be handled in a more automated system, for instance:

“partner” means a spouse, civil partner or one of a couple whether of the same sex or opposite sex who although not married to each other have *been registered in the National Register as sharing the same accommodation for at least two years, or are registered as parents to a common child or common children and are registered in the National Register as sharing the same accommodation.* (new elements are italicised)

Although terrible prose (!), the example demonstrates that in automated government systems, legal conditions must be of a formal nature obviating human evaluation. Official entries in government registers and government decisions in individual cases could be employed instead. I am not claiming that the two new conditions included in the example above would be politically acceptable. The point is merely that politicians in the legislative process should be aware of the requirements entailed in automated government administration and should take these requirements into account when making legal substantive choices. If legislation is formulated in ways which presuppose individual scrutiny of how people are “living together” and if they “treat each other as spouses”, while at the same time trying to meet demands for cheaper and more automated government procedures, it is likely that the implemented result will not be in line with the legislators’ intentions. In democratic systems, the lawmakers’ intentions should always be decisive. Legislators should thus be assisted to establish optimum, well-founded trade-offs between political fairness and effective administrative routines.

It may very well be that politicians choose the level of individual treatment and scrutiny following from the first example given above, thereby both accepting that automation is not possible or desirable, and that extra costs linked to individual case processing will accrue. Of course, it follows from democratic principles that legislators are free to make such political choices. However, combining old-fashioned regulatory technique and government policy with a strong emphasis on modernisation and automated administrative work may well result in incompatible political goals. In the view of this author, conflicting goals for future computerised exercise of government powers should be addressed and solved in an open, political process. Desire to avoid minutiae should not make legislators refrain from involving

themselves in questions concerning significant legal substantive choices and the design of tomorrow's government administration.

Publicatio legis

In 304 BC, the secretary of the Pontifical Council, Gnaeus Flavius, succeeded in making public a codification of technical rules in the law of Rome (*legis actiones*). The primary law of the Twelve Tables was publicly known, but great uncertainty existed as to what constituted the proper forms, actions and wording, as well as to how the law should be understood. Part of the problem was very casuistic provisions which were applied analogically pursuant to the secret technical rules of the Council. For instance, could provisions on the liability of the owner of a carriage pulled by a horse be applied to owners of a carriage pulled by a bull? Before Gnaeus Flavius published the technical rules, the pontiffs had a virtual monopoly on the law in that they were the only ones who knew, and could selectively disseminate their legal knowledge of proper law and valid interpretations.²⁵

Publication of *legis actiones* gave the plebeians access to legal predictability. Under the rule of law and in democracies of law, we are about to develop a huge volume of very detailed rules which describe the proper understanding and procedures of traditionally designed legislation. This new legal understory growth represents pronouncements from government agencies which are to a large extent accepted without contradiction, but are not visible or accessible to ordinary men and women. In democracies and under the rule of law, we must avoid obscure legislation, and systems development project groups must thus not be allowed to develop into covert lawmakers. Even the law of our computerised society should always remain in the public domain.

Notes

¹ Organised as a project internally in the agency, or by inviting software developers to make systems according to requirements specified by the agency.

² The main knowledge base of this article is case studies in (Schartum 1993: 356-435) and interviews regarding the relationship between development of government decision systems and legislative drafting (Schartum & Eide 2016, forthcoming in the CompLex series).

³ In some areas of law, calculations of benefits and taxes etc. will dominate, while trying of conditions is the most important in other areas, for instance in connection with school and university admission processes. Sergot (1991: 20) makes a distinction between implicit and explicit representation of law with "algorithmic programs", "data processing application" and "pay roll systems" as examples of the first category. To this author's knowledge, implicit representation clearly is the dominating way of representing law in public administration, particularly within "mass administration", cf. below.

⁴ Cormacain (2013: 5), analyses the nature of legislation which may make it difficult to access.

⁵ Other models are also possible; for instance that private software firms develop systems according to a standard requirement specification issued by the government agency.

⁶ Hjelseth (2012: 4) makes distinctions between three converting levels: “Transcribe” indicates that a regulatory statement may be directly processed automatically; “transform” reflects that transcription requires further assessment; and “transfer” means that automatic processing is impossible and that the rules must be transferred to professional (manual) interpretation. Here, I treat transcribe and transform cases as one, as questions of transformation.

⁷ Many government agencies, or system owners, do not have sufficient in-house personnel with the required technological and legal competencies. Thus, external software developers are often engaged to carry out large parts of the job, under contract and supervision of the government agency in question.

⁸ Others concern e.g. technological questions, questions of reorganisation, financial questions, graphical design and interface etc. Schartum (2010) emphasise the close connections between development of technological systems, relevant regulations and reorganisation of work tasks.

⁹ One of few in-depth analyses of the legal contents in programming code of government agencies, could be found in Schartum (1994: 190-232).

¹⁰ e.g. regarding money, hours, age etc.

¹¹ Instead, discretionary rules may e.g. be referred to a manual process, or as a yes or no input to whether or not a discretionary condition is satisfied. Grimmelmann (2005: 1732) talks of rules and standards, and underlines that standards invite decisionmakers to exercise discretion and may thus not be automated.

¹² Some exceptions may be made e.g. due to particularly high costs of transformation, low practical significance, or if it is expected that relevant provisions will be amended or repealed.

¹³ Grimmelmann (2005: 1740) compares software with physical architecture and laws of nature.

¹⁴ For a historical survey, see Bing (1977: 196) about research concerning automated legal decisions and “automation-friendly” legislation.

¹⁵ Taxation of individual tax payers and processing of applications to the State Educational Loan Fund are examples of government areas where percentage of fully automated cases is very high.

¹⁶ See Hildonen and Gulstuen (2012: 27).

¹⁷ Pasquale (2015: 8) sees “The Secret Judgements of Software” as a general problem, not only relevant to government sector: “So why does this all matter? It matters because authority is increasingly expressed algorithmically. Software encodes thousands of rules and instructions computed in a fraction of a second.”

¹⁸ See survey of different uses of the black box metaphor in Wikipedia (2016).

¹⁹ Specification also takes the form of information/data models and process models, etc.

²⁰ Besides, programming code is frequently machine generated in rule engines, i.e. a digital tool that automatically translates semi-formal specifications to code that can be run in a computer system. A description of a rule engine and how it works is found in Eriksen and Smogeli (2007: 23).

²¹ Taylor and Bench-Capon (1991: 95 - 113) discuss how knowledge based systems developed for the legal domain could be designed to support legislators, e.g. regarding solution specification.

²² See Fiedler (1973). Kennedy (2016: 82) suggests “E-regulation” as new field of study. Although most elements of what could be encompassed by such a field are

known and well established, it may be fruitful to analyse all these aspects in integrated ways; thereby also creating a broader understanding and more interest.

²³ It is however, important to stress that formulating laws and at the same time taking care to special considerations regarding automated decision-making, does not imply writing laws in programming style. Rather, characteristics will be logically stringent texts, concise and consistent use of concepts etc. – all expressed in natural language.

²⁴ However, in most cases computer programs will be legally correct and thus represent valid law. In Norway, government decision-making systems are normally quasi-binding, i.e. although they only express the government view on how the law should be interpreted they have an actual binding effect.

²⁵ Based on Bing (2016), and Schiller (1978: §55, §133).

References

- Bing, J. (1977). Automatiseringsvennlig lovgivning. *Tidsskrift for Rettsvitenskap*: 195-229.
- Bing, J. (2016). *Fremtidens fortid*. Oslo, Gyldendal forlag (forthcoming).
- Cormacain, R. (2013). Have the Renton Committee's Recommendations on Electronic Access to Legislation been Fulfilled? *European Journal of Current Legal Issues* 19(3) Available at: <http://webjcli.org/article/view/264/361>
- Eriksen, Ø. and Smogeli A. L. (2007). *Rule Engine* (Master thesis, Norwegian University of Science and Technology, Department of Computer and Information Science), Trondheim: Norwegian University of Science and Technology. Available at: http://brage.bibsys.no/xmlui/bitstream/handle/11250/252837/536428_FULLTEXT01.pdf?sequence=3&isAllowed=y
- Fiedler, H. (1973). Computer-conscious law-making. In *Data Processing in the Government*, Luxemburg. Available at the Norwegian Research Center for Computers and Law (NRCCL) collection of conference papers, Oslo, Norway.
- Grimmelmann, J. (2005). Regulation by Software. *Yale Law Journal*, Vol. 114: 1719-58.
- Hildonen, B.H. and Gulstuen, S. (2012). *Kartlegging av automatiserte avgjørelser i offentlig forvaltning*. Oslo: Datatilsynet 2012. Available at: http://www.uio.no/studier/emner/jus/afin/FINF4001/h12/undervisningsmateriale/automatisert_forvaltning.pdf
- Hjelseth, E. (2012). Experiences on converting interpretative regulations into computable rules, *Proceedings of the CIB W78 2012: 29th International Conference – Beirut, Lebanon*.
- Kennedy, R. (2016). E-regulation and the rule of law: Smart government, institutional information infrastructures, and fundamental values. *Information Polity* 21(1): 77-98. <http://dx.doi.org/10.3233/IP-150368>
- National Assembly for Wales. The Well-being of Future Generations (Wales). (Act 2015 (Registrable Interests) Regulations 2015, Laid before the National Assembly for Wales 2 November 2015, Part I.) Available at: <http://www.legislation.gov.uk/anaw/2015/2/contents/enacted>
- Pasquale, F. (2015). *The Black Box Society. The Secret Algorithms That Control Money and Information*, Harvard University Press. <http://dx.doi.org/10.4159/harvard.9780674736061>

- Schartum, D.W. & Eide, C.S. (2016). *Forprosjekt vedrørende digitalisering av bygningsretten* (Kartleggingprosjektet). Oslo, forthcoming in CompLex.
- Schartum, D.W. (2010). Developing eGovernment Systems – Legal, Technological and Organizational Aspects, *Scandinavian Studies in Law* (56): 125–147.
- Schartum, D.W. (1994). Dirt in the Machinery of Government – Legal Challenges Connected to Computerized Case Processing in Public Administration. *International Journal of Law and Information Technology* 94(2): 327–354. <http://dx.doi.org/10.1093/ijlit/2.3.327>
- Schartum, D.W. (1993). *Rettsikkerhet og systemutvikling i offentlig forvaltning*. Oslo, Universitetsforlaget.
- Schiller, A. A. (1978). *Roman law: mechanisms of development*. The Hague; New York: Mouton. <http://dx.doi.org/10.1515/9783110807196>
- Sergot, M. (1991). Chapter 1 - The Representation of Law in Computer Programs. In Bench-Capon, T.J.M. (ed.) *Knowledge-Based Systems and Legal Applications* (Vol. 36, pp. 3-67). Academic Press, London. <http://dx.doi.org/10.1016/b978-0-12-086441-6.50006-4>
- Taylor, A. and Bench-Capon, T.J.M. (1991). Support for the formulation of legislation. In Bench-Capon, T.J.M. (ed.) *Knowledge-Based Systems and Legal Applications* (Vol. 36, pp. 95-113). Academic Press, London 1991. <http://dx.doi.org/10.1016/B978-0-12-086441-6.50010-6>
- Black Box. (n.d). In *Wikipedia*. Retrieved March 1, 2016, from https://en.wikipedia.org/wiki/Black_box