C. Procedures Reconceived and Replaced for the Automated State

1. Protecting Individual Rights

Automation jeopardizes the due process guarantees of meaningful notice and opportunity to be heard. Both technological and legal mechanisms can secure meaningful notice, combat automation bias, and enhance the accuracy of decisions about constitutionally significant individual rights.

a. Securing Meaningful Notice

The inadequacy of notice in this automated age must be addressed.[367] At a minimum, automated systems should generate audit trails that record the facts and rules supporting their decisions.[368] Audit trails should include a comprehensive history of decisions made in a case, including the identity of the individuals who recorded the facts and their assessment of those facts.[369] Audit trails should detail the actual rules applied in every mini-decision that the system makes. With audit trails, agencies would have the means to provide individuals with the reasons supporting an automated system's adjudication of their important rights.

Requiring audit trails would adopt a norm that other countries, such as Australia, have embraced.[370] It would ensure that agencies uniformly provide detailed notice to individuals, no matter the identity of the private vendor that built the system. A per se requirement would guard against a contractor's inclination to omit this feature as a cost-saving measure. It would facilitate judicial review of the different factual adjudications made by human operators.

Furthermore, audit trails might help to combat a hearing officer's presumption that automated decisions are infallible. By providing a detailed map of a computer's decision-making process, audit trails would

[367] In the "No Fly" context, national security concerns may outweigh an individual's right to advanced notice of her "No Fly" status. But in nearly all other instances, courts should insist on strict compliance with notice requirements.

[368] *See* AUS. ADM. REV., AUTOMATED ASSISTANCE, *supra* note 69, at 23; WEITZNER ET AL., *supra* note 197, at 4–5 (describing TAMI software architecture that records justifications for data matching and mining "No Fly" determinations to ensure transparency in such systems, including clear view of logical, factual, and legal bases for inferences made by system).

[369] AUS. ADM. REV., AUTOMATED ASSISTANCE, *supra* note 69, at 23, 48. This can be accomplished by additional prompts that ask operators to describe their thought processes. Some jurisdictions, such as Australia, already provide such prompts to operators. *Id.*

[370] *Id.* at 46–47.

encourage officers to critically assess the computer's specific findings. Although developing the capability to produce audit trails would have initial fixed costs, it would have few variable ones.

Individual agencies would naturally determine the precise contours of such audit trails and the notices they generate. Agencies should ensure that their audit trails follow industry best practices.[371] They, and reviewing courts, should apply the familiar *Mullane* standard to ensure that the system provides sufficient notice.[372] This approach would encourage agencies to incorporate advances in the design of audit trails into their systems.

b. Protections for Hearings

Administrative law must confront the automation bias that threatens the impartiality of hearing officers and deprives individuals of meaningful opportunities to be heard.[373] This may be more difficult than avoiding hearing officers' biases for or against particular parties because recusal is unlikely to help. Two rules could be employed to combat such bias.

First, agencies should make it clear to hearing officers that automated systems are fallible.[374] To that end, hearing officers should receive explicit training about the phenomenon of automation bias.[375] Studies demonstrate that individuals who receive such training are more likely to scrutinize an automated system's suggestions.[376] The training of judges has been effective in a parallel enterprise. Special workshops on scientific theory and methodology have provided needed training to federal district court judges charged with assessing the reliability of expert testimony.[377]

[371] A rule prescribing the exact contours of an audit trail would surely be inadvisable given the rapidity of change in technology and industry best practices.

[372] *See supra* text accompanying notes 204–06 (discussing due process notice formulation of Mullane v. Central Hanover Bank & Trust Co., 339 U.S. 306, 319 (1950)).

[373] *See supra* notes 150–54 and accompanying text (discussing the phenomenon of automation bias and its risks).

[374] *See* John K. Hawley, *Patriot Fratricides: The Human Dimension Lessons of Operation Iraqi Freedom*, FIELD ARTILLERY, Jan.–Feb. 2006, at 18, *available at* http://sill-www.army.mil/famag/2006/JAN_FEB_2006/JAN_FEB_06_PAGES_18_19.pdf (arguing that soldiers operating automated aircraft systems need to be aware that computerized system is fallible so that soldiers can provide meaningful check on system's decisions).

[375] *See* Skitka et al., *supra* note 151, at 94 (arguing that automation bias can be dispelled by training decision makers about phenomenon).

[376] *Id.*; *cf.* Neil M. Richards, *The Information Privacy Law Project*, 94 GEO. L.J. 1087, 1098–99 (2006) (drawing parallels between movement in antidiscrimination law to redesign workplace design to combat implicit cognitive bias and the response in information privacy scholarship to implicit cognitive limitations of actors in information transactions).

[377] Scott Brewer, *Scientific Expert Testimony and Intellectual Due Process*, 107 YALE L.J. 1535,

Second, agencies should require hearing officers to explain, in detail, their reliance on an automated system's decision.[378] Officers should identify the computer-generated facts or legal findings on which they relied in making their decisions. This accords with administrative law's longstanding faith in the prophylactic power of requiring explicit statements of reasons. Asking hearing officers to evaluate the basis for their decisions would further mitigate the effects of automation bias.[379]

Fighting automation bias is a logical next step from *Daubert v. Merrell Dow Pharmaceuticals, Inc.*[380] and *Kumho Tire Co. v. Carmichael.*[381] In those cases, the Court upheld the exclusion of unreliable expert testimony due to the jury's inability to reject evidence carrying the imprimatur of science or engineering.[382] If the Seventh Amendment countenances taking issues away from juries due to the risks of scientific bias,[383] surely administrative law can similarly address automation bias. These suggested requirements are certainly less invasive than the exclusion of evidence in *Daubert* and *Kumho*. Requiring warnings and statements of reasons would be the equivalent of cautionary instructions, not flat bans on officers beset by automation bias.

Providing a means to combat the effects of automation bias would not undermine the Court's traditional reluctance to dissect the motives of decision makers. In *United States v. Morgan*,[384] an agency head publicly announced his disappointment about the Court's reversal of an earlier decision of his agency.[385] Addressing allegations of bias, the Court refused to assess the administrator's convictions on matters of policy.[386] The Court reasoned that decision makers are presumed to be persons of "conscience and intellectual discipline."[387]

The reporting and warning requirements would not transgress the letter or the spirit of *Morgan*. They would not entail an individual assessment of

1677 (1998).

[378] Cummings, *supra* note 147, at 4.

[379] *Id.* (requiring operators to explain extent to which they relied on automation leads to fewer instances of bias).

[380] 509 U.S. 579 (1993).

[381] 526 U.S. 137 (1999).

[382] *Daubert*, 509 U.S. at 597; *Kumho*, 526 U.S. at 158; *see* ERICA BEECHER-MONAS, EVALUATING SCIENTIFIC EVIDENCE: AN INTERDISCIPLINARY FRAMEWORK FOR INTELLECTUAL DUE PROCESS 232 (2007) (arguing that judicial gatekeeping of scientific evidence is matter of due process and fairness due to risk of overreliance on junk science by juries).

[383] BEECHER-MONAS, *supra* note 382, at 232.

[384] 313 U.S. 409 (1941).

[385] *Id.* at 420.

[386] *Id.* at 420–21.

[387] *Id.* at 421.

a particular hearing officer's inclination to adopt computerized decisions. Instead, they would erect safeguards to dispel the influence of automation bias without intruding on and evaluating a particular judge's thought process.

Reconceptualizing the *Mathews* balancing test would also help individuals to receive meaningful hearings. As suggested in Part II,[388] the *Mathews* analysis needs a more realistic way to compare the fixed costs of corrective action with the future benefits of that correction.[389] The theoretical foundations of *Mathews* would, in turn, reflect the new realities of the automated administrative state. It might countenance requiring additional costs in cases where retrofitting an automated system's reasoning was essential to enabling individuals to address an agency's intended actions.

For instance, a reconceived *Mathews* test might permit hearings on flaws in the CBMS software in a case involving a child whose Medicaid benefits were abruptly terminated. Because CBMS does not generate audit trails, and because automation bias will likely influence the hearing officer, expert testimony would be critical to demonstrate that the computer decision is flawed. If experts discover a distortion in the encoded policy that, once fixed, would avoid errors in thousands of other cases, the *Mathews* analysis might support such additional process. In the "No Fly" context, if independent advisory boards were provided testing data that included false positives, the system might be recalibrated to be more accurate. Coupled with measures to combat automation bias, this approach would vindicate due process values for the automated administrative state.

2. *Replacing Rulemaking Procedures*

Automated systems must be designed with transparency and accountability as their primary objectives, so as to prevent inadvertent and procedurally defective rulemaking. This approach incorporates several basic norms of behavior.

First, vendors should release systems' source codes to the public.[390] Opening up the source code would reveal how a system works.[391] It would

[388] *See supra* text accompanying notes 243–53.

[389] *See supra* text accompanying notes 248–50.

[390] *See* Citron, *supra* note 32 (arguing that the release of various information systems' source codes used by agencies, including decision-making systems, would advance various models of the administrative state); Lawrence Lessig, *Open Code and Open Societies*, *in* PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE 358 (Joseph Feller et al. eds., 2005).

shed light on the policies encoded in it.[392] The Office of Management and Budget could issue a circular conditioning the provision of federal funding for technology purchases on the use of open code.[393] A state budget office could do the same for local purchases receiving state aid.[394] Alternatively, legislators could mandate open code systems.[395] Certain systems such as the "No Fly" program, however, might fall outside an open code mandate because public safety concerns might outweigh transparency's benefits.[396]

One might argue that the public's ability to identify encoded policy changes would force agencies to engage in notice-and-comment rulemaking, the costs and delays of which have already caused it to fall into disuse. This concern, however, is inapplicable in cases where the public has identified programming mistakes that can be fixed. For example, a software glitch may have caused CBMS to apply illegal income and asset requirements to breast cancer patients on Medicaid.[397] There, the agency programmers may have understood established policy, but nonetheless executed it improperly. An open code system would help ensure the correction of encoded policy without involving expensive rulemaking procedures.

This argument, however, does indeed have force in cases where programmers' interpretations establish new rules. CBMS's inquiry about whether individuals seeking food stamps were "beggars" arguably falls into that category.[398] Programmers' translations of policy into code inevitably involve some interpretation, and it would be impossible to force

[391] David M. Berry & Giles Moss, *Free and Open-Source Software: Opening and Democratising e-Government's Black Box*, 11 INFO. POLITY 21, 23 (2006).

[392] Dr. John Henry Clippinger, Senior Fellow at the Berkman Center for Internet & Society at The Harvard Law School and Director of the Open Identity Meta-system, suggests that programmers write the code in natural languages, which lend themselves to policy commands and are easier to read. Comments of Dr. John Henry Clippinger, Podcast, Berkman Center for Internet and Society at Harvard Law School, "Danielle Citron on *Technological Due Process*," Jan. 15, 2008, http://blogs.law.harvard.edu/mediaberkman/2008/01/15/danielle-citron-on-technological-due-process-podcast/.

[393] Citron, *supra* note 32.

[394] *Id.*

[395] *See* Jyh-An Lee, *New Perspectives on Public Goods Production: Policy Implications of Open Source Software*, 9 VANDERBILT J. OF ENT. & TECH. L. 45, 61 (2006).

[396] Citron, *supra* note 32.

[397] *See supra* note 133 and accompanying text (discussing programming error that imposed income and asset limits on breast cancer patients on Medicaid in violation of federal and state law).

[398] *See supra* note 217 and accompanying text (discussing CBMS inquiry into applicant's status as a "beggar"). Agency programmers seemingly did not generate notes documenting their thought processes while building the decision tables. My FOIA request for the decision tables and accompanying notes only yielded the decision table fixes that I cite throughout the Article. Such notes, however, would be fruitful in ascertaining whether the question "Are you a beggar?" can be attributed to policy interpretations made by programmers or bureaucrats.

agencies to write rules on every possible interpretation of a rule. But certain interpretations, such as the "beggar" inquiry, arguably exceed what would constitute a permissible interpretation and instead arbitrarily create new rules that deserve notice-and-comment rulemaking.[399] If so, an agency should consult with programmers before publishing its notice of proposed rulemaking in order to clearly establish the precise rule that would be automated.

Second, agencies should be required to test a system's software.[400] Agencies should maintain testing suites that run expected and unexpected hypothetical scenarios designed by independent policy experts through decision systems to expose distorted policy.[401] Agencies should invest resources into designing such suites, which should include many complex and varied cases with expected outcomes.[402]

Testing protocols should be run before a system's launch, during implementation, and every time policies change. Federal procurement regulations could require contracts to specify that decision systems pass testing suites before states can accept systems from vendors.[403] This would prevent programmers who otherwise might test a system from foregoing testing in the face of dwindling resources.[404]

Rigorous testing reflects a norm of proper software development.[405] Testing would help identify and eliminate a programmer's bias.[406] It would

[399] *See, e.g.*, Hoctor v. U.S. Dep't of Agric., 82 F.3d 165, 169–72 (7th Cir. 1996) (striking down interpretative rule imposing eight-foot fence requirement as a procedurally defective rule that required notice-and-comment because interpretation was arbitrary choice that did not necessarily derive from the rule).

[400] AUS. ADM. REV., AUTOMATED ASSISTANCE, *supra* note 69, at 40, 45; PETE GOODLIFFE, CODE CRAFT: THE PRACTICE OF WRITING EXCELLENT CODE 130 (2007); Mark Underseth, *A Test-Driven Approach to Developing Embedded Software*, EVALUATION ENGINEERING, Apr. 2007, at 44. Testing is a term of art that includes a variety of methodologies. GOODLIFFE, *supra*, at 130. Naturally, the particular type of testing that should be used would be determined by systems specialists who could identify the "best practices" for testing a particular system.

[401] *See* Comments of Dr. John Henry Clippinger, *supra* note 392. Although outcome testing cannot cover every possible situation, the Australian government notes that it would test enough scenarios for agencies to have a high level of comfort in the rules embedded in the system. AUS. ADM. REV., AUTOMATED ASSISTANCE, *supra* note 69, at 45.

[402] To address policy changes and possible leaks to vendors, the contents of the testing suite should be constantly evolving and expanding.

[403] I thank my colleague David Super for this insightful point.

[404] *See* Terry Shepard, Margaret Lamb, & Diane Kelly, *More Testing Should be Taught*, 44 COMM. OF THE ACM, June 2001, at 103, 104 (explaining that because testing is usually allocated at end of software development process, it bears brunt of resource constraints). It may be worth considering whether the details of testing protocols should be kept confidential to prevent vendors from "teaching" to the test by inserting hidden work-arounds into the system so that it produces the right answers in the testing suite.

[405] AUS. ADM. REV., AUTOMATED ASSISTANCE, *supra* note 69, at 21; GOODLIFFE, *supra* note 400, at 130 (explaining that testing is central to building software).

pinpoint policies that have been altered in their translation to code.[407] Testing would identify software bugs that distort policy. Although testing would not provide all of the benefits of obtaining public comment on proposed rules, it would provide a critical layer of assurance about the content of encoded policy and the algorithms employed by systems.[408]

To be sure, testing would entail fixed costs. But it would add negligible variable ones. It is certainly cheaper, and arguably more precise, to find defects in a computer program's decisions before its implementation than it is to uncover ex post a software program's flaws in individual cases.

Despite the fact that rigorous testing is a baseline requirement of software development, independent contractors did not adequately test CBMS or TIERS before those systems went live.[409] This caused many of the problems that currently plague those systems. Absent a testing requirement, contractors seeking to submit the lowest bid are unlikely to build testing costs into their project budgets. As state and federal agencies automate more of their decisions, often relying upon many of the same private contractors,[410] a uniform testing rule is essential.

Trial-run testing embodies the notion that the accountability of rulemaking can be replaced with an adjudication involving interested parties. In *NLRB v. Bell Aerospace Co.*,[411] the Court allowed the agency to make policy through such adjudications in lieu of formal or informal rulemaking.[412] The Court explained that adjudicative procedures can produce "relevant information necessary to mature and fair consideration of the issues" in the same way that rulemaking generates informed views of affected groups.[413]

Testing would serve as a substitute, albeit an imperfect one, for the interested parties of *Bell Aerospace*. Testing provides a set of hypothetical scenarios to reveal software's flaws.[414] Although broad public

[406] Friedman & Nissenbaum, *supra* note 77, at 344 (arguing in favor of rules regarding testing of software to detect programmer bias).

[407] GOODLIFFE, *supra* note 400, at 130; Underseth, *supra* note 400, at 44.

[408] *See* Grimmelmann, *supra* note 29, at 1738 ("The ambition of software engineering is to remove the most important bugs and reduce the unpredictable errors (i.e., 'wrong' decisions) to a tolerable level, not to eliminate them entirely.").

[409] *See supra* notes 139, 145–47 (discussing involvement of EDS and Accenture in building CBMS and TIERS).

[410] *See supra* note 65 (detailing role that EDS will play in building future Medicaid automated systems).

[411] 416 U.S. 267 (1974).

[412] *Id.* at 295.

[413] *Id.*

[414] Dr. John Henry Clippinger and Gene Koo, Senior Fellow and Fellow, respectively, at the Berkman Center on the Internet and Society at Harvard Law School, both suggested that vendors

participation naturally would not occur in testing regimes, the set of scenarios designed to expose a system's errors would serve as a surrogate for clients adjudicating cases, which would reveal a software flaw.[415]

Third, agencies should explore ways to allow the public to participate in the building of automated decision systems. Such participation is not without precedent.[416] For instance, states and localities have established welfare reform boards that create opportunities for the public to participate in setting welfare policy.[417] The City of Denver has a Welfare Reform Board, consisting of appointees of the Mayor, representatives of the business community, service providers, and welfare recipients.[418]

In the same vein, agencies could establish information technology review boards that would provide opportunities for stakeholders and the public at large to comment on a system's design and testing. Although finding the ideal makeup and duties of such boards would require some experimentation, they would secure opportunities for interested groups to comment on the construction of automated systems that would have an enormous impact on their communities once operational.

Last, agencies might consider refraining from automating policy that has not undergone formal or informal rulemaking procedures, such as interpretative rules and policy statements. The public and often elected officials have not had the opportunity to respond to these policies.[419] Programmers encoding interpretative rules and policy statements are simply too far removed from the democratic process to justify the significant risk of distorted policy that automation entails.[420] Although adherence to rulemaking procedures would not solve the accountability

establish testing suites before a system's official launch. *See* Comments of Dr. John Henry Clippinger and Gene Koo, *supra* note 392. Bureaucrats and interested stakeholders should be permitted to submit unusual scenarios for testing. *Id.* The testing of such scenarios would allow vendors to identify problems that plague these systems. *Id.*

[415] Surrogate advocacy plays an important role in administrative law. Because of principles against retroactive application of new policies and doctrines keeping alive mooted disputes capable of repetition yet evading review, some litigants in landmark cases before administrative agencies—and courts—actually have little immediate stake in the outcome of their litigation.

[416] Diller, *supra* note 294, at 1213 (giving examples of such public participation in welfare administration).

[417] *Id.*

[418] *Id.*

[419] 5 U.S.C. § 553(b)(A) (2000) (interpretative rules and policy statements are not subject to notice-and-comment requirement); *see supra* notes 285–86 (defining interpretative rules and policy statements).

[420] Policy statements also should not be automated because such statements should not be designed to control rights and duties of individuals, which is precisely what automation would accomplish. *See* Anthony, *supra* note 277, at 1315 (arguing that agencies are not entitled to make policy statements binding).

deficit accompanying today's inadvertent delegation of legislative power to code writers, it would allay some of these concerns.