Journal of the American Medical Informatics Association, 25(5), 2018, 496–506
doi: 10.1093/jamia/ocx106
Advance Access Publication Date: 16 October 2017
Research and Applications

Research and Applications

Clinical decision support alert malfunctions: analysis and empirically derived taxonomy

Adam Wright,[1,2,3] Angela Ai,[1] Joan Ash,[4] Jane F Wiesen,[4] Thu-Trang T Hickman,[1] Skye Aaron,[1] Dustin McEvoy,[3] Shane Borkowsky,[5] Pavithra I Dissanayake,[6] Peter Embi,[7] William Galanter,[8] Jeremy Harper,[9] Steve Z Kassakian,[4] Rachel Ramoni,[10,11] Richard Schreiber,[12] Anwar Sirajuddin,[13] David W Bates,[1,2,3] and Dean F Sittig[14]

[1]Department of Medicine, Brigham and Women's Hospital, Boston, MA, USA, [2]Department of Medicine, Harvard Medical School, Boston, MA, USA, [3]Clinical and Quality Analysis, Partners Healthcare, Somerville, MA, USA, [4]Department of Medical Informatics and Clinical Epidemiology, Oregon Health and Science University, Portland, OR, USA, [5]Department of Medicine, University of Illinois at Chicago, Chicago, IL, USA, [6]Department of Pathology, University of Illinois at Chicago, Chicago, IL, USA, [7]Regenstrief Institute, Indianapolis, IN, USA, [8]Department of Medicine, Pharmacy Practices, and Pharmacy Systems, Outcomes and Policy, University of Illinois at Chicago, Chicago, IL, USA, [9]Department of Biomedical Informatics, The Ohio State University Wexner Medical Center, Columbus, OH, USA, [10]Department of Oral Health Policy and Epidemiology, Harvard School of Dental Medicine, Boston, MA, USA, [11]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA, [12]Department of Medicine and Information Technology, Holy Spirit Hospital – A Geisinger Affiliate, Camp Hill, PA, USA, [13]Department of Medical Informatics, Memorial Hermann Health System, Houston, TX, USA and [14]Department of Biomedical Informatics, University of Texas Health Science Center at Houston, TX, USA

Corresponding Author: Adam Wright, Brigham and Women's Hospital and Harvard Medical School, 1620 Tremont St., Boston, MA 02115, USA. E-mail: awright@bwh.harvard.edu. Phone: (617) 525-9811. Fax: (617) 732-7072

Received 23 February 2017; Revised 7 August 2017; Editorial Decision 28 August 2017; Accepted 2 September 2017

ABSTRACT

Objective: To develop an empirically derived taxonomy of clinical decision support (CDS) alert malfunctions.

Materials and Methods: We identified CDS alert malfunctions using a mix of qualitative and quantitative methods: (1) site visits with interviews of chief medical informatics officers, CDS developers, clinical leaders, and CDS end users; (2) surveys of chief medical informatics officers; (3) analysis of CDS firing rates; and (4) analysis of CDS overrides. We used a multi-round, manual, iterative card sort to develop a multi-axial, empirically derived taxonomy of CDS malfunctions.

Results: We analyzed 68 CDS alert malfunction cases from 14 sites across the United States with diverse electronic health record systems. Four primary axes emerged: the cause of the malfunction, its mode of discovery, when it began, and how it affected rule firing. Build errors, conceptualization errors, and the introduction of new concepts or terms were the most frequent causes. User reports were the predominant mode of discovery. Many malfunctions within our database caused rules to fire for patients for whom they should not have (false positives), but the reverse (false negatives) was also common.

Discussion: Across organizations and electronic health record systems, similar malfunction patterns recurred. Challenges included updates to code sets and values, software issues at the time of system upgrades, difficulties with migration of CDS content between computing environments, and the challenge of correctly conceptualizing and building CDS.

Conclusion: CDS alert malfunctions are frequent. The empirically derived taxonomy formalizes the common recurring issues that cause these malfunctions, helping CDS developers anticipate and prevent CDS malfunctions before they occur or detect and resolve them expediently.

Key words: clinical decision support, electronic health records, safety, anomaly detection

BACKGROUND

Well-designed clinical decision support (CDS) systems embedded in electronic health record (EHR) systems have been repeatedly shown to improve the quality and safety of health care.[1–11] Use of CDS, especially in relation to medications, is widespread and includes examples such as drug interaction checking, drug-allergy alerts, and medication dosing alerts.[12,13] In the United States, CDS adoption has been spurred, at least in part, by the federal Meaningful Use incentive program, which mandated several forms of CDS.[14–17]

However, developing and maintaining an effective CDS system is challenging. Guidelines, regulations, and medical evidence must be interpreted and translated into a form that can be executed in the EHR;[18–24] this transformation often requires resolving ambiguity and developing consensus. Over time, knowledge assets accumulate and must be both managed and maintained. The tools, processes, and resources needed for knowledge management can be complex, costly, and personnel- and time-intensive.[25–34] Governance of CDS and clinical knowledge bases can also be challenging, particularly in large organizations.[34–38]

One particularly pernicious manifestation of the complexity of CDS systems is their potential for malfunction. We define a CDS malfunction as an event where a CDS intervention does not work as designed or expected. This broad definition is designed to encompass a range of possible issues, and is perhaps best understood through an example we recently described. In that case, there was a malfunction of a CDS rule that suggested thyroid testing in patients who had been taking amiodarone for at least 1 year.[39] The internal ID number for amiodarone had been changed in the EHR's medication dictionary, but the alert logic was not updated to account for this new code. Thereafter, the alert stopped firing for patients newly started on the drug.

According to a recent survey we conducted, 93% of chief medical information officers (CMIOs) have experienced CDS malfunctions at their organizations, and 65% reported experiencing, on average, at least 1 malfunction a year.[39]

CDS malfunctions exist in a broader context of EHR safety and EHR system problems. Seminal literature in this field includes work by Koppel on the role that computerized physician order entry (CPOE) systems can play in facilitating medication errors,[40] and by Westbrook on the "manifestation, mechanisms, and rates" of errors in 2 CPOE systems in Australia.[41] Development of taxonomies has been especially useful for increasing our understanding of errors. For example, Amato developed a taxonomy of medication errors related to CPOE based on an analysis of safety reports filed in the United States.[42] Magrabi et al.[43] developed a 32-category taxonomy of computer patient safety incidents in Australia and subsequently expanded it to 36 categories by reviewing data from the US Food and Drug Administration.[44] Sittig et al.[45] likewise developed a taxonomy of health information technology–related safety concerns, and Kushnirik et al.[46] conducted an international comparison of efforts to improve health information technology–related patient safety in Canada, the United States, and England.

A range of methods have been used to create taxonomies.[47–49] Many taxonomies are created de novo by experts, while others are derived empirically. Some taxonomies are designed to be collectively exhaustive (ie, to categorize all possible items in a set), while others are left open, particularly in the common situation where not all possible members of a set to be categorized have yet been observed (or even occurred, in situations where the set under study is evolving).

Our research group has previously applied empirical taxonomy development methods to EHRs and to CDS. For example, we used qualitative methods to identify unintended consequences of EHR implementation and organized the myriad consequences reported into a taxonomy of 9 major categories.[50] In the area of CDS, we developed a taxonomy of the functions of CDS systems (such as data elements used, actions, and choices offered) by reviewing a large number of CDS artifacts from a large integrated delivery network.[51] Similarly, we developed a taxonomy of front-end tools in commercial and internally developed EHR systems by reviewing the features of 53 front-end CDS tools with the help of 11 informants.[52]

Although these taxonomies of health information technology safety issues and CDS system capabilities are very useful, neither set of taxonomies considers safety issues for CDS, a prevalent issue based on the findings of our survey. We recently published a case series of 4 CDS malfunctions[39] from a single organization with a single EHR. Beyond our cases series, only scattered case reports of CDS malfunctions exist in the literature,[39,53–55] even though our survey results suggest that they occur frequently.[39]

Because no systematic analysis of CDS malfunctions has been published, nor has any classification of such malfunctions been developed, we aimed to close these gaps by collecting a large number of reports of CDS malfunctions from a range of institutions and developing a new, empirically derived, 2-layer taxonomy of the causes, modes of discovery, and manifestations of CDS malfunctions.

METHODS

Data collection

To ensure that our empirically derived taxonomy was broad and representative, we developed a large database of CDS malfunctions. We used both qualitative and quantitative methods to collect data, including site visits with interviews, surveys of experts, and analyses of CDS firing and override logs.

The largest number of cases came from a series of interviews we conducted at 5 health care organizations across the United States (Figure 1). At the 5 sites, we conducted a total of 92 interviews with knowledge engineers, CDS analysts and developers, clinical champions, CDS committee members, CMIOs, and end users. We conducted semistructured interviews, first asking general questions about CDS development and use, knowledge management, and governance and monitoring. We then asked each respondent to describe specific CDS malfunctions he or she had encountered, using the definition above and providing examples from our published case report as needed to stimulate recall. All interviews were audio-recorded and transcribed. In total, the interviews produced 50.8 h of audio and >2500 typed pages of transcripts, which were thoroughly analyzed and coded using NVivo (QSR International, Melbourne, Australia). Specific examples of malfunctions were tagged and extracted into our malfunction database. In many cases, several informants at a particular organization described the same malfunction. When this occurred, we combined the details of all of the accounts to get a more complete picture and asked for additional details from the sites in the case of conflicting information or gaps. Many of the issues described in the interviews were found during testing and were

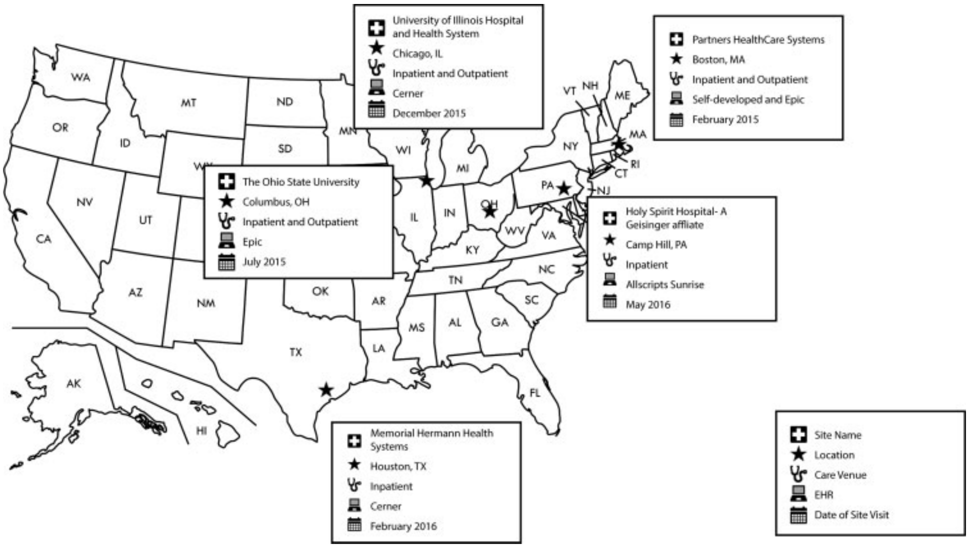


Figure 1. Site visit and interview locations.

corrected before CDS went live in the production system; these cases were not included in our database unless they were related to a software defect in the EHR.

To supplement the interviews, we also used an online survey form to solicit additional case reports (https://goo.gl/o1klE2). We solicited case reports through the American Medical Informatics Association Clinical Information Systems and CDS working group electronic mailing lists, the Association of Medical Directors of Information Systems electronic mailing list, and the electronic mailing list of the Scottsdale Institute, a collaborative member organization focused on sharing information and best practices about health information technology. The case report was anonymous, though submitters had the option to provide their contact information if they were willing to be re-contacted about the case(s) they submitted.

Because our interview and survey methods were only able to collect data on CDS malfunction cases that were already known to our informants, we also conducted 2 additional investigations to identify previously unknown anomalies: one analyzing alert firing data, and another analyzing override comments. In our analysis of alert firing data, 4 organizations provided us with deidentified data for 5 EHRs (one site had recently switched EHRs and provided data for its new and old systems). We received data from 2 Epic systems (Verona, WI, USA), 1 Cerner system (Kansas City, MO, USA), 1 Allscripts Sunrise system (Atlanta, GA, USA), and 1 self-developed system. The data included the number of times each alert fired each day for a period of at least 1 year. We then visualized the data to identify apparent step changes, spikes, and dips in firing. When such anomalies were identified, we worked with the site to investigate the cause. In some cases, the apparent step changes, spikes, and dips were intentional (eg, an alert that was permanently or seasonally disabled), but in several cases, they represented previously unknown malfunctions. In our analysis of override reason comments, we identified additional malfunctions from one site that provided override reason comment data entered by users. All of these cases were added to our malfunction database. The flow of cases is shown in Figure 2.

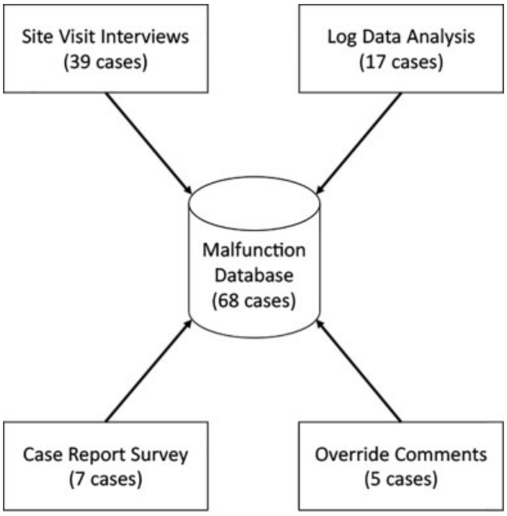
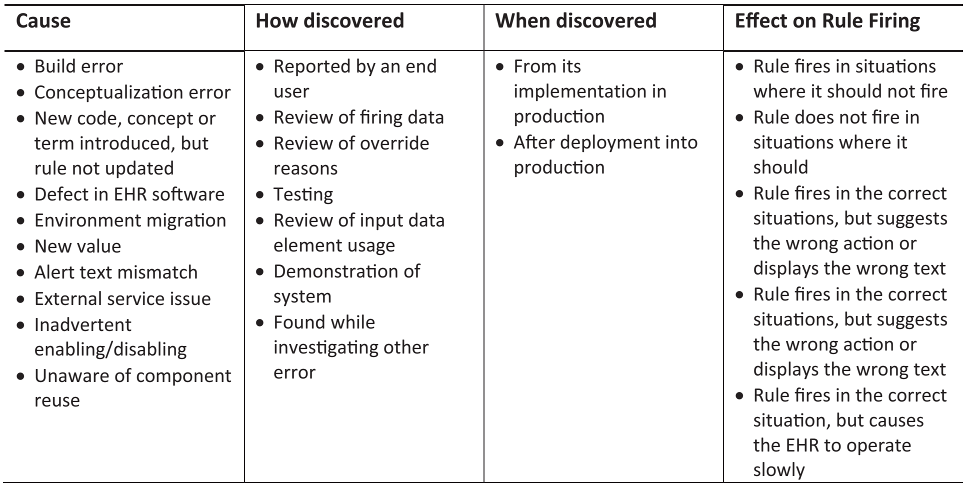


Figure 2. Flow of cases.

We stored all of the CDS malfunction cases in a secure Research Electronic Data Capture[56] database hosted on a server behind the Partners Healthcare System firewall. All study procedures were reviewed and approved by the Partners Healthcare System Human Subjects Committee, and by each institution's local Institutional Review Board.

Empirical taxonomy derivation

The empirically derived taxonomy was developed iteratively, beginning after the first site visit when the site visit team (AW, JA, AA, and DFS) reviewed the cases identified by interviewees, added them to a database, and did a preliminary categorization of them. The taxonomy, which was initially one-dimensional, was further developed after each additional site visit. At the point when we felt we knew enough about malfunctions to write survey questions, we developed and conducted the electronic survey. Cases found through the other methods (survey, firing analysis, and override analysis) were also added to the database. After the database of CDS



Cause	How discovered	When discovered	Effect on Rule Firing
• Build error • Conceptualization error • New code, concept or term introduced, but rule not updated • Defect in EHR software • Environment migration • New value • Alert text mismatch • External service issue • Inadvertent enabling/disabling • Unaware of component reuse	• Reported by an end user • Review of firing data • Review of override reasons • Testing • Review of input data element usage • Demonstration of system • Found while investigating other error	• From its implementation in production • After deployment into production	• Rule fires in situations where it should not fire • Rule does not fire in situations where it should • Rule fires in the correct situations, but suggests the wrong action or displays the wrong text • Rule fires in the correct situations, but suggests the wrong action or displays the wrong text • Rule fires in the correct situation, but causes the EHR to operate slowly

Figure 3. Overview of CDS malfunctions taxonomy.

malfunctions was completed, the research team reviewed all the cases it contained and conducted a series of manual card sorts[47,49,57] to develop a more detailed multi-axial empirically derived taxonomy of CDS malfunctions. We previously used the same card-sort technique for other empirical taxonomy derivation exercises related to CDS and EHRs.[50–52,58–61] Although semi-automated methods for taxonomy development exist, in our experience they are not as robust as expert curation when the number of cases is feasible to review, as it was in this case. The card sort was completed over 3 rounds, including 2 synchronous sessions (totaling 7.5 h). As the taxa emerged, AA and AW also conducted repeated review and reclassification of each case, as needed, and at times in consultation with case contributors, with a final review by all of the taxonomy developers.

The taxonomy developed is empirical: it is derived from a collection of real-world CDS malfunctions that were reported by diverse organizations. There are potentially other axes or taxa that were not uncovered during our analysis; as such, the empirically derived taxonomy is meant to be practically useful and grounded in real-world experience, but it is not exhaustive, and can and should be extended as more CDS malfunctions come to light.

Further, the precise definitions and bounds of the concept of taxonomy and related concepts, such as ontology, hierarchy, dictionary, and controlled vocabulary, are often debated in informatics. For this paper, we adopt the plain-language meaning of taxonomy from the Oxford English Dictionary: “a classification of something; a particular system of classification.” Our empirically derived taxonomy has 4 axes, with a number of classifications (taxa) in each axis. Each taxon has a defining name and examples, as well as empirically derived frequency information. However, the empirically derived taxonomy is not intended to be exhaustive (it is an open-world taxonomy), and there is no higher-level network of relationships between or among the taxa, except for the hierarchy implicit in the 4 axes.

RESULTS

Malfunction cases collected

Across the 4 methods, we gathered a total of 82 malfunction cases. After review, we excluded 14 cases, 9 because insufficient details were available about the cases to adequately classify them and 5 because they appeared to describe issues unrelated to CDS. All further results are derived from the 68 cases that could be analyzed. Supplementary Appendix S1 contains descriptions and categorizations of each of the 68 cases; in some cases, details of the case were removed to preserve the anonymity of the organization where the case occurred.

The 68 cases came from a total of 14 clinical sites across the United States: 8 academic medical centers, 4 large integrated delivery systems, and 2 community hospitals. The majority of sites reporting cases used Epic, Cerner, or Allscripts systems (likely reflecting the general usage of EHRs at similar sites in the United States), as well as several self-developed systems.

Empirically derived taxonomy

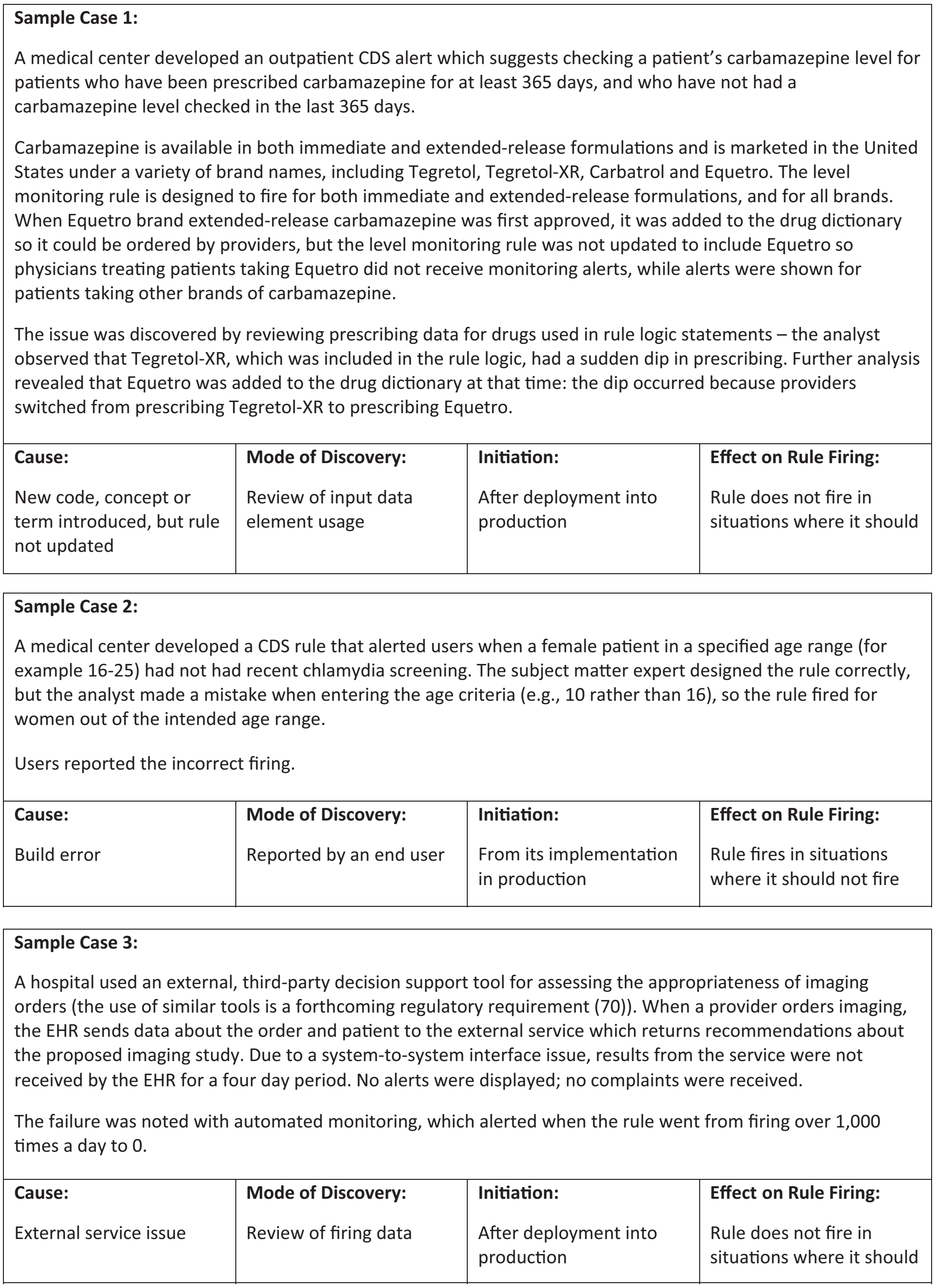
After completing the card sort, 4 primary axes emerged: the cause of the malfunction, the mode of discovery of the malfunction, whether the malfunction was present from the initial deployment of the CDS or developed later, and how the malfunction affected rule firing. Additional axes were considered, such as the clinical impact of the malfunction and the element or component of the CDS system that was involved; however, insufficient data were available regarding the first, and the second proved to be quite specific to each EHR system. The entire empirically derived taxonomy is shown in Figure 3.

To illustrate the type of cases identified and the axes of our empirically derived taxonomy, we present 3 sample cases in Figure 4. Each case also includes the 4 taxonomy terms that were applied to the case. The 4 axes of the empirically derived taxonomy are detailed in the next 4 sections, as well as the 4 related tables.

Axis 1: causes

Table 1 shows the causes of the CDS malfunctions. The most common were build errors, which we defined as errors that occur when a rule is designed correctly but built incorrectly (eg, if an analyst accidentally enters a wrong value for an age criterion while building a rule). Also common were conceptualization errors, which occur when a rule is not designed correctly (eg, an important inclusion or exclusion criterion is omitted during the design of a rule). Conceptualization errors can occur when the rule designer either fails to consider a possible issue or properly considers an issue but fails to record it in the rule specification.

The next most common cause of malfunctions was the release of new codes, particularly new medication and laboratory codes. In many cases, medication and laboratory dictionaries are under the



Sample Case 1:

A medical center developed an outpatient CDS alert which suggests checking a patient's carbamazepine level for patients who have been prescribed carbamazepine for at least 365 days, and who have not had a carbamazepine level checked in the last 365 days.

Carbamazepine is available in both immediate and extended-release formulations and is marketed in the United States under a variety of brand names, including Tegretol, Tegretol-XR, Carbatrol and Equetro. The level monitoring rule is designed to fire for both immediate and extended-release formulations, and for all brands. When Equetro brand extended-release carbamazepine was first approved, it was added to the drug dictionary so it could be ordered by providers, but the level monitoring rule was not updated to include Equetro so physicians treating patients taking Equetro did not receive monitoring alerts, while alerts were shown for patients taking other brands of carbamazepine.

The issue was discovered by reviewing prescribing data for drugs used in rule logic statements – the analyst observed that Tegretol-XR, which was included in the rule logic, had a sudden dip in prescribing. Further analysis revealed that Equetro was added to the drug dictionary at that time: the dip occurred because providers switched from prescribing Tegretol-XR to prescribing Equetro.

Cause:	**Mode of Discovery:**	**Initiation:**	**Effect on Rule Firing:**
New code, concept or term introduced, but rule not updated	Review of input data element usage	After deployment into production	Rule does not fire in situations where it should

Sample Case 2:

A medical center developed a CDS rule that alerted users when a female patient in a specified age range (for example 16-25) had not had recent chlamydia screening. The subject matter expert designed the rule correctly, but the analyst made a mistake when entering the age criteria (e.g., 10 rather than 16), so the rule fired for women out of the intended age range.

Users reported the incorrect firing.

Cause:	**Mode of Discovery:**	**Initiation:**	**Effect on Rule Firing:**
Build error	Reported by an end user	From its implementation in production	Rule fires in situations where it should not fire

Sample Case 3:

A hospital used an external, third-party decision support tool for assessing the appropriateness of imaging orders (the use of similar tools is a forthcoming regulatory requirement (70)). When a provider orders imaging, the EHR sends data about the order and patient to the external service which returns recommendations about the proposed imaging study. Due to a system-to-system interface issue, results from the service were not received by the EHR for a four day period. No alerts were displayed; no complaints were received.

The failure was noted with automated monitoring, which alerted when the rule went from firing over 1,000 times a day to 0.

Cause:	**Mode of Discovery:**	**Initiation:**	**Effect on Rule Firing:**
External service issue	Review of firing data	After deployment into production	Rule does not fire in situations where it should

Figure 4. Sample cases with taxonomy coding applied.

control of the pharmacy and the laboratory, respectively. Organizations frequently encountered issues when these codes were changed but rules that depended on them were not modified to match. This issue manifested in both false negatives (eg, if the pharmacy changes the code for a medication that triggers a drug monitoring alert and the alert stops firing) and false positives (if the laboratory changes the code for the monitoring test, the alert may continue to fire even if the test is done). Defects in EHR software (which required either a "patch" from the EHR vendor or a workaround by local CDS implementers) were also common, and almost always manifested during system upgrades.

Another common cause of malfunctions was computer environment migration. Most organizations had at least 3 computing environments or domains for their EHR: one for developing CDS (often called "DEV"), one for testing CDS (often called "TEST"), and the main production EHR used to provide care for patients (often called "PROD"). During normal CDS development, the content is developed in the DEV environment, migrated (either manually or with a utility) to TEST, and finally deployed in PROD. Issues where CDS logic was corrupted or did not work as expected during migration were common (eg, an analyst might migrate a rule to PROD but not the related medication value sets), as were situations where CDS worked in TEST

Table 1. Axis 1: What caused the malfunction? (n = 68)

Cause	No. of cases	Description	Example
Build error	14	The clinical conceptualization and design of the rule were sound, but there was an error building the rule that caused it to malfunction.	An alert was designed to fire for patients using inhaled fluticasone propionate. The rule was specified properly, but during the build, no route limitation was configured, so the alert also fired inappropriately for patients using fluticasone propionate nasal spray.
Conceptualization error	11	There was an error in the clinical conceptualization of the rule, which, when implemented, led to a malfunction.	A rule is designed to fire for patients receiving all heparins, but the subject matter expert only includes unfractionated heparin codes in the rule specification. Because of this oversight, the alert does not fire for patients receiving low-molecular-weight heparin, which should have been included.
New code, concept, or term introduced, but rule not updated	10	A new item was added to one of the data dictionaries used to encode clinical concepts within the EHR's database.	A new extended-release form of carbamazepine is added to the EHR's drug dictionary (other extended-release forms were already in the dictionary). A rule is in place that suggests monitoring carbamazepine levels. It identifies patients taking carbamazepine by looking to see whether one of several specified carbamazepine codes is on the patient's medication list. The list of codes in the rule was not updated, so the rule did not fire for patients taking the new form of the medication.
Defect in EHR software	10	There is a programming error in the source code of the EHR that causes the EHR to function other than as designed or documented, and this error causes CDS to malfunction.	The EHR has a routine for importing and exporting rules. A bug in the importing routine causes certain diagnosis codes to be scrambled during the process.
Environment migration	7	Moving the rules from one environment to another (eg, test to production) leads to a malfunction.	A rule currently being tested, but not ready for deployment, is accidentally moved into the production environment when another completely tested package of CDS content is moved.
New value	6	The set of allowable values or the meaning of the values of a coded concept is changed.	The unit of measure for internal representation of weight is changed from pounds to kilograms, causing rules that use weight in calculations to fire incorrectly.
Alert text mismatch	3	The text displayed by the rule and the logic of the rule do not match.	An alert states, "Patient has CAD-equivalent on problem list and a beta blocker is not on the medication list. Recommend beta blocker." However, it also fires for patients who do not have coronary artery disease (CAD) on the problem list but have had at least one visit with CAD listed as an encounter diagnosis.
External service issue	4	An external software routine that is used in CDS either fails to function or cannot be reached, creating an error.	An external drug classification service stops working (ie, always returns "false") after maintenance, causing an alert that suggests aspirin for patients with CAD who are not already taking aspirin or another anti-platelet drug to fire for all patients who have CAD, even if they are already taking aspirin.
Inadvertent enabling/disabling	2	A rule in the production environment is inappropriately enabled or disabled.	A rule suggests influenza vaccination for patients who have not yet received it. The rule is supposed to be manually enabled every year when vaccine is available and disabled at the end of flu season. In 2 separate years, the staff member responsible for enabling the alert was on leave and the alert was never enabled.
Unaware of component reuse	1	A "building block" for CDS, such as a criteria specification or a value set, is reused in multiple rules and a knowledge engineer modifies it, intending to change one rule, but inadvertently also causes a change in another rule that uses the same component.	A coded value set that defines coronary artery disease is used in multiple rules. The value set is modified to correct an issue in one rule, but the change causes another rule to malfunction, as the knowledge engineer was unaware that the same value set was used in multiple rules.

but malfunctioned in PROD due to configuration differences between the environments. Also frequent was the inadvertent migration of rules into PROD before they had been completely tested.

Related to new codes, changes in expected values for the codes were also a frequent problem. For example, a rule might expect a numeric potassium result, such as 4.0 mEq/L, but malfunction when a result such as "specimen hemolyzed" was present. Issues with values above or below assay ranges (eg, "$\geq$ 60" or "not detected") or unexpected units of measure (weights stored internally in kilograms rather than pounds) also occurred. Less common issues were failures in external computing services that CDS relied on (eg, a medication classification service), mismatches between the logic of the alert and the displayed text of the alert, inadvertent enabling or disabling of a rule by a knowledge engineer, and modification of a component of one rule that caused an inadvertent change in another rule.

Axis 2: mode of discovery

Table 2 describes the modes of malfunction discovery for the 68 cases analyzed. By far, the most common mode of discovery for CDS malfunctions was user reporting. Users were much more likely to notice and report false positive alerts (ie, incorrect alerts) rather than false negatives, where an alert should have fired but did not.[39] Although some user reports were made by "special people"[62] like CMIOs or clinical champions, most appeared to be spontaneous reports by typical end users. A variety of reporting channels were used, including opening help desk tickets, filing safety reports, or contacting knowledge engineers and CMIOs directly.

Axis 3: start of malfunction

As shown in Table 3, just over half of the cases were "congenitally" defective, malfunctioning as soon as they were deployed. In the other cases, the CDS initially worked, but stopped working at some point due to some external perturbation such as a software upgrade or a change in a code or value in another system; these may be harder to detect. For the cases where a rule stopped working, there was often a substantial delay from when the malfunction began to when it was detected, reported, and corrected.

Table 2. Axis 2: How was the malfunction discovered? (n = 68)

Mode of Discovery	No. of cases	Description	Example/Notes
Reported by an end user	37	A user notifies the help desk, files a safety report, or directly contacts the team responsible for CDS.	A user calls the help desk because an alert that suggests mammography is showing for every patient, even young children and men.
Review of firing data	17	Retrospective examination of system-generated alert firing logs reveals anomalies (unusual patterns, such as large spikes or no firings for a long period of time).	A data analyst notices that an alert that discourages repeat echocardiography has not fired in the last year when it previously fired several times a day.
Review of override reasons	5	Retrospective examination of user-entered alert override reasons was conducted.	A researcher notices that an alert that suggests insulin for patients in the emergency department is frequently overridden with comments like "Patient already received insulin." Further investigation reveals a logic error that causes the alert to miss certain prior insulin administrations.
Testing	5	Testing of CDS revealed malfunctions.	An analyst testing in the production environment in advance of a go-live at a new hospital joining an existing system notices that a sepsis alert does not fire when it should.
Review of input data element usage	2	Changes in the pattern of use of data elements that are used by the rule logic (eg, orderable items or lab result) were reviewed.	An analyst observes that a new thyroid-stimulating hormone (TSH) test result code is being used at one site in an integrated system and knows that a drug monitoring alert depends on TSH results, and that the rule has not been updated to include the new result code.
Demonstration of system	1	An error was identified while demonstrating how the system worked.	A researcher demonstrating a drug-lab interaction monitoring CDS could not get an alert to fire when it should have.
Found while investigating other error	1	Malfunction in one rule was found incidentally while investigating an error in another rule.	A knowledge engineer investigating a particular CDS problem incidentally observes that an alert related to renal dose adjustment is also not firing correctly.

Table 3. Axis 3: When did the malfunction start? (n = 68)

Initiation	No. cases	Description	Example
From its implementation in production	37	Rule never fired correctly in production environment.	A new rule is released that suggests pneumococcal vaccination. Due to a configuration problem, it only fired for pediatric patients when it was intended to fire for patients over the age of 18 years.
After deployment into production	31	Rule initially worked correctly but began to fail at some later point.	A hospital laboratory starts reporting TSH results under a new internal code number when it changes to a next-generation assay (the code is changed from "TSH" to "TSH3"). An alert is in place that suggests TSH testing for patients who are eligible and have not recently had the test. When the code is changed, the alert starts firing for patients who have recently had the new test. The alerts had previously worked correctly.

Table 4. Axis 4: What was the effect of the malfunction on rule firing? (n = 68)

Effect	No. of cases	Description	Example
Rule fires in situations where it should not fire	33	Rule fires for patients who do not meet the intended criteria.	After a software upgrade, a pregnancy alert starts firing for women of all ages, even those who are very unlikely to be pregnant and who are intended to be excluded from the rule logic.
Rule does not fire in situations where it should	18	Rule fails to fire for all patients, or a subset who do, in fact, meet the intended criteria.	The logic for an alert that suggests lead screening for 2-year-old children is inadvertently modified by a knowledge engineer and the rule stops firing entirely.
Rule fires in the correct situations, but suggests the wrong action or displays the wrong text	14	The rule fires for the correct patients, but the display text of the rule, or the actions that it offers, are incongruent with its purpose.	A rule that suggests adding sickle cell disease to the problem list is able to be overridden if the patient does not have sickle cell disease. Due to an issue with rule configuration, the acknowledgment reason offered is "Patient does not have gestational diabetes" rather than "Patient does not have problem."
Rule fires in the correct situation, but causes the EHR to operate slowly	3	The EHR operates more slowly due to an issue with the way the CDS logic is constructed (commonly because it fetches too much data or depends on an external service that is slow).	An alert that suggests adding hypertension to the problem list for patients with serial high blood pressure measurements queries the EHR's observation table in an inefficient way. In certain situations, this causes a processing backlog that slows all aspects of the EHR.

Axis 4: effect on rule firing

Table 4 shows the effect of malfunctions on the rate of rule firing. When analyzing the impact of CDS malfunctions on firing, a range of patterns was observed. In certain cases (wrong rule action or system slowness related to a rule), the rule continued to fire at the correct times, so there was no effect on the pattern of rule firing.

DISCUSSION

During the course of analyzing the 68 malfunction cases used to develop this empirically derived taxonomy, the number of repeating themes was striking. Across organizations and EHR systems, similar malfunction patterns recurred, including challenges at the time of system upgrades, maintaining consistent code sets and values, difficulties with environment migration, and the challenge of correctly designing and implementing CDS.

Although we anticipated many of the most commonly observed causes of CDS malfunctions through our experience and review of the literature,[39,53–55] we did not anticipate the relatively large number of cases related to migration between environments. We believe that better tools and testing procedures for migration of CDS content are needed. Particularly problematic is that several organizations reported that they had to rebuild CDS in each environment by hand, leading to frequent discrepancies in rules between environments. Some organizations also reported a policy against testing in production, which makes it difficult to ensure that rules are actually working once they reach the production system, and also difficult to troubleshoot issues that appear in production but not in the test environment.[63]

A concerning pattern was the majority of CDS malfunctions being reported by end users. We believe that a robust testing and monitoring strategy should be able to prevent most CDS malfunctions before they reach users and patients, or at least detect them far sooner than users typically report them. Most organizations reported an approach for pre-implementation testing; fewer reported post–go-live testing or monitoring. We believe that both pre-implementation testing and post–go-live testing and monitoring need to be improved to ensure more reliable CDS.

The common causes of CDS malfunctions that we identified lend themselves to a variety of potential solutions. For example, build errors were prevalent – better testing, build reviews, and easier-to-use build tools are likely to help reduce the prevalence of build errors. Conceptualization errors have diverse causes, but clear formats for preparing specifications (in our experience, flowcharts work particularly well), design reviews, and multidisciplinary design teams are likely to be helpful. Issues related to new or changed terms were also common, and better communication processes between those responsible for management of terms and concepts and those responsible for CDS are essential for preventing these errors; more robust knowledge management tools and processes can also be useful.

One emerging issue was CDS malfunctions related to issues with external services;[20,22,27,64–76] this problem is likely to grow with time. Some CDS systems now rely on external vocabulary and classification systems, or even entirely upon external rule engines. When these external systems malfunction, CDS can misfire. These issues manifested as false positive alerts, false negative alerts, and system slowness (sometimes extending beyond CDS to the entire EHR). Although these external systems are promising for their positive impact on scalability and maintainability of CDS knowledge, additional research and development is needed to ensure the reliability of these external systems and optimize their performance, particularly since several vendors are likely to be involved in their development and maintenance.[77]

An important question related to CDS malfunctions is: Who is responsible for CDS maintenance? This question becomes especially complex in the case where some content is acquired from an EHR vendor or a third-party content provider. Although we believe that provider organizations are ultimately responsible for their CDS content, we also believe that EHR vendors and content suppliers should provide tools for monitoring CDS in real time (or as near to real time as possible) and consider enhancements to their content-authoring and knowledge-management tools to prevent the common issues observed in our study (eg, better environment migration tools to ensure content integrity, or automated dependency checking tools to mitigate potential issues when new codes are changed).

As described in Methods, our taxonomy is empirically derived and thus grounded in real-world experience. However, it is not exhaustive. Reporting of the cases was voluntary and mostly spontaneous (though some cases were found through detailed log analysis). As such, there are likely additional cases of CDS malfunctions that we did not find; it is possible that some of these cases may not neatly fit the empirically derived taxonomy we proposed. We encourage CDS developers and researchers to report these cases through our online submission form (https://goo.gl/o1klE2) and to consider preparing them as case reports for publication; better and more frequent publication and discussion of CDS malfunctions is almost certainly helpful for increasing our understanding of them, as well as developing tools and procedures for detecting or, better yet, preventing them. Although our empirically derived taxonomy is meant to be practically useful and, given the diversity of participating organizations and data collection methods, likely covers the most common taxa in each axis, it is also open for extension and expansion as more cases become available.

Our study has several strengths, including the mixed-methods approach, the geographic and technical diversity of the sites, and the deep analysis of the malfunction cases. Furthermore, the largest previously reported published case series of CDS malfunctions consisted of only 4 reports, and thus this new analysis of 68 cases is, by far, the largest of its kind. However, our study also has some important limitations. The nonexhaustive nature of the empirically derived taxonomy is a key limitation. Further, all of the sites that participated in the study were in the United States; it is possible that additional issues might be seen in other countries, though there was no appreciable geographic variation in case patterns across the United States, and many of the vendor systems used by our US-based informants are also used worldwide.

CONCLUSION

CDS malfunctions are widespread, but the same patterns of malfunctions seem to recur consistently across organizations and EHR vendors. A better understanding of these patterns is essential for detecting and preventing CDS malfunctions. Likewise, improved tools and processes are needed to ensure the reliability of CDS systems.

CONTRIBUTORS

AW had full access to all the data in the study and takes responsibility for the integrity of the data and accuracy of the data analysis. AW, JA, DWB, and DS are responsible for study concept and design. All authors participated in data acquisition, interpretation, and analysis. AW drafted the manuscript. All authors provided critical revisions of the manuscript for important intellectual content.

FUNDING AND COMPETING INTERESTS

The research reported in this publication was supported by the National Library of Medicine of the National Institutes of Health under award number R01LM011966. The content of this article is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors do not have any competing interests.

ACKNOWLEDGMENT

We acknowledge and appreciate the contributions of the interview subjects and survey respondents who provided the cases we analyzed in this study.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

REFERENCES

1. Sequist TD, Gandhi TK, Karson AS, *et al*. A randomized trial of electronic clinical reminders to improve quality of care for diabetes and coronary artery disease. *J Am Med Inform Assoc*. 2005;12(4):431–37.
2. Bright TJ, Wong A, Dhurjati R, *et al*. Effect of clinical decision-support systems: a systematic review. *Ann Int Med*. 2012;157(1):29–43.
3. Wolfstadt JI, Gurwitz JH, Field TS, *et al*. The effect of computerized physician order entry with clinical decision support on the rates of adverse drug events: a systematic review. *J General Int Med*. 2008;23(4):451–58.
4. Pearson S-A, Moxey A, Robertson J, *et al*. Do computerised clinical decision support systems for prescribing change practice? A systematic review of the literature (1990–2007). *BMC Health Services Res*. 2009;9(1):1.
5. Kawamoto K, Houlihan CA, Balas EA, Lobach DF. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ*. 2005;330(7494):765.
6. Kaushal R, Shojania KG, Bates DW. Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Arch Int Med*. 2003;163(12):1409–16.
7. Jaspers MW, Smeulers M, Vermeulen H, Peute LW. Effects of clinical decision-support systems on practitioner performance and patient outcomes: a synthesis of high-quality systematic review findings. *J Am Med Inform Assoc*. 2011;18(3):327–34.
8. Hunt DL, Haynes RB, Hanna SE, Smith K. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review. *JAMA*. 1998;280(15):1339–46.
9. Garg AX, Adhikari NK, McDonald H, *et al*. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA*. 2005;293(10):1223–38.
10. Chaudhry B, Wang J, Wu S, *et al*. Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. *Ann Int Med*. 2006;144(10):742–52.
11. Roshanov PS, Fernandes N, Wilczynski JM, *et al*. Features of effective computerised clinical decision support systems: meta-regression of 162 randomised trials. *BMJ*. 2013;346:f657.
12. Hsiao C-J, Hing E, Socey TC, Cai B. *Electronic Medical Record/Electronic Health Record Systems of Office-based Physicians: United States, 2009 and Preliminary 2010 State Estimates*. Atlanta: National Center for Health Statistics; 2010.
13. Hsiao C-J, Hing E. *Use and Characteristics of Electronic Health Record Systems Among Office-Based Physician Practices, United States, 2001–2012*. Atlanta: National Center for Health Statistics; 2012.
14. Marcotte L, Seidman J, Trudel K, *et al*. Achieving meaningful use of health information technology: a guide for physicians to the EHR incentive programs. *Arch Int Med*. 2012;172(9):731–36.
15. Jha AK. Meaningful use of electronic health records: the road ahead. *JAMA*. 2010;304(15):1709–10.
16. Ip IK, Schneider LI, Hanson R, *et al*. Adoption and meaningful use of computerized physician order entry with an integrated clinical decision support system for radiology: ten-year analysis in an urban teaching hospital. *J Am College Radiol*. 2012;9(2):129–36.
17. Blumenthal D, Tavenner M. The "meaningful use" regulation for electronic health records. *New Engl J Med*. 2010;363(6):501–04.
18. Wang D, Peleg M, Tu SW, *et al*. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. *Int J Med Inform*. 2002;68(1):59–70.
19. Boxwala AA, Tu S, Peleg M, *et al*. Toward a representation format for sharable clinical guidelines. *J Biomed Inform*. 2001;34(3):157–69.
20. Sittig DF, Wright A, Osheroff JA, *et al*. Grand challenges in clinical decision support. *J Biomed Inform*. 2008;41(2):387–92.

21. McCoy A, Wright A, Eysenbach G, *et al*. State of the art in clinical informatics: evidence and examples. *Yearb Med Inform*. 2013;8(1):13–19.
22. Paterno MD, Maviglia SM, Ramelson HG, *et al*. Creating shareable decision support services: an interdisciplinary challenge. *AMIA Annu Symp Proc*. 2010.
23. Kim M, Turechek Z, Boxwala A, *et al*. Definition of a metadata model for a multi-layered clinical practice guideline representation framework. *Int J Funct Inform Personalised Med*. 2012;4(1):47–55.
24. Boxwala AA, Rocha BH, Maviglia S, *et al*. A multi-layered framework for disseminating knowledge for computer-based decision support. *J Am Med Inform Assoc*. 2011;18 (Suppl 1):i132–i39.
25. Sittig DF, Wright A, Simonaitis L, *et al*. The state of the art in clinical knowledge management: an inventory of tools and techniques. *Int J Med Inform*. 2010;79(1):44–57.
26. Kakabadse NK, Kakabadse A, Kouzmin A. Reviewing the knowledge management literature: towards a taxonomy. *J Knowledge Manag*. 2003;7(4):75–91.
27. Dixon BE, Simonaitis L, Goldberg HS, *et al*. A pilot study of distributed knowledge management and clinical decision support in the cloud. *Artif Intell Med*. 2013;59(1):45–53.
28. Bali RK. *Clinical Knowledge Management: Opportunities and Challenges*. Hershey, PA: IGI Global; 2005.
29. Alavi M, Leidner DE. Review: knowledge management and knowledge management systems: conceptual foundations and research issues. *MIS Quarterly*. 2001;25(1):107–36.
30. Sittig DF, Wright A, Meltzer S, *et al*. Comparison of clinical knowledge management capabilities of commercially-available and leading internally-developed electronic health records. *BMC Med Inform Decis Mak*. 2011;11(1):1.
31. Rocha RA, Bradshaw RL, Bigelow SM, *et al*. Towards ubiquitous peer review strategies to sustain and enhance a clinical knowledge management framework. AMIA Annu Symp Proc. 2006:654–58.
32. Glaser J, Hongsermeier T. Managing the investment in clinical decision support. In: *Clinical Decision Support*. Burlington, MA: Elsevier; 2006.
33. Ash JS, Sittig DF, Guappone KP, *et al*. Recommended practices for computerized clinical decision support and knowledge management in community settings: a qualitative study. *BMC Med Inform Decis Mak*. 2012;12(1):1.
34. Wright A, Bates DW, Middleton B, *et al*. Creating and sharing clinical decision support content with Web 2.0: issues and examples. *J Biomed Inform*. 2009;42(2):334–46.
35. Wright A, Sittig DF, Ash JS, *et al*. Governance for clinical decision support: case studies and recommended practices from leading institutions. *J Am Med Inform Assoc*. 2011;18(2):187–94.
36. Sirajuddin AM, Osheroff JA, Sittig DF, Chuo J, Velasco F, Collins DA. Implementation pearls from a new guidebook on improving medication use and outcomes with clinical decision support: effective CDS is essential for addressing healthcare performance improvement imperatives. *J Healthcare Inform Manag*. 2009;23(4):38.
37. Osheroff JA. *Improving Medication Use and Outcomes with Clinical Decision Support: A Step by Step Guide*. Chicago: HIMSS; 2009.
38. Lyman JA, Cohn WF, Bloomrosen M, Detmer DE. Clinical decision support: progress and opportunities. *J Am Med Inform Assoc*. 2010;17(5):487–92.
39. Wright A, Hickman TT, McEvoy D, *et al*. Analysis of clinical decision support system malfunctions: a case series and survey. *J Am Med Inform Assoc*. 2016;23(6):1068–76.
40. Koppel R, Metlay JP, Cohen A, *et al*. Role of computerized physician order entry systems in facilitating medication errors. *JAMA*. 2005;293(10):1197–203.
41. Westbrook JI, Baysari MT, Li L, Burke R, Richardson KL, Day RO. The safety of electronic prescribing: manifestations, mechanisms, and rates of system-related errors associated with two commercial systems in hospitals. *J Am Med Inform Assoc*. 2013;20(6):1159–67.
42. Amato MG, Salazar A, Hickman TT, *et al*. Computerized prescriber order entry–related patient safety reports: analysis of 2522 medication errors. *J Am Med Inform Assoc*. 2017;24(2):316–22.
43. Magrabi F, Ong MS, Runciman W, Coiera E. An analysis of computer-related patient safety incidents to inform the development of a classification. *J Am Med Inform Assoc*. 2010;17(6):663–70.
44. Magrabi F, Ong MS, Runciman W, Coiera E. Using FDA reports to inform a classification for health information technology safety problems. *J Am Med Inform Assoc*. 2012;19(1):45–53.
45. Sittig DF, Classen DC, Singh H. Patient safety goals for the proposed Federal Health Information Technology Safety Center. *J Am Med Inform Assoc*. 2015;22(2):472–78.
46. Kushniruk AW, Bates DW, Bainbridge M, Househ MS, Borycki EM. National efforts to improve health information system safety in Canada, the United States of America and England. *Int J Med Inf*. 2013;82(5):e149–60.
47. Hedden H. *The Accidental Taxonomist*. 2nd ed. Medford, NJ: Information Today; 2016.
48. Rosenfeld L, Morville P, Arango J. *Information Architecture: for the Web and Beyond*. 4th ed. Sebastopol, CA: O'Reilly Media; 2015.
49. Abbas J. *Structures for Organizing Knowledge: Exploring Taxonomies, Ontologies, and Other Schemas*. New York: Neal-Schuman Publishers; 2010.
50. Campbell EM, Sittig DF, Ash JS, Guappone KP, Dykstra RH. Types of unintended consequences related to computerized provider order entry. *J Am Med Inform Assoc*. 2006;13(5):547–56.
51. Wright A, Goldberg H, Hongsermeier T, Middleton B. A description and functional taxonomy of rule-based decision support content at a large integrated delivery network. *J Am Med Inform Assoc*. 2007;14(4):489–96.
52. Wright A, Sittig DF, Ash JS, *et al*. Development and evaluation of a comprehensive clinical decision support taxonomy: comparison of front-end tools in commercial and internally developed electronic health record systems. *J Am Med Inform Assoc*. 2011;18(3):232–42.
53. Ray S, Wright A. Detecting anomalies in alert firing within clinical decision support systems using anomaly/outlier detection techniques. *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM; 2016: 185–90.
54. Chen ES, Borlawsky T, Qureshi K, Li J, Lussier YA, Hripcsak G. Monitoring the function and use of a clinical decision support system. *AMIA Annu Symp Proc*. 2007:902.
55. Hripcsak G. Monitoring the monitor: automated statistical tracking of a clinical event monitor. *Comput Biomed Res*. 1993;26(5):449–66.
56. Harris PA, Taylor R, Thielke R, Payne J, Gonzalez N, Conde JG. Research electronic data capture (REDCap): a metadata-driven methodology and workflow process for providing translational research informatics support. *J Biomed Inform*. 2009;42(2):377–81.
57. Lincoln YS, Guba EG. *Naturalistic Inquiry*. Beverly Hills, CA: Sage Publications; 1985.
58. Dykstra RH, Ash JS, Campbell E, *et al*. Persistent paper: the myth of "going paperless." *AMIA Annu Symp Proc*. 2009:158.
59. Campbell EM, Guappone KP, Sittig DF, Dykstra RH, Ash JS. Computerized provider order entry adoption: implications for clinical workflow. *J General Int Med*. 2009;24(1):21–26.
60. Ash JS, Sittig DF, Dykstra R, Campbell E, Guappone K. The unintended consequences of computerized provider order entry: findings from a mixed methods exploration. *Int J Med Inform*. 2009;78:S69–S76.
61. Ash JS, Sittig DF, Dykstra R, Campbell E, Guappone K. Exploring the unintended consequences of computerized physician order entry. *Stud Health Technol Inform*. 2007;129(1):198.
62. Wright A, Ash JS, Erickson JL, *et al*. A qualitative study of the activities performed by people involved in clinical decision support: recommended practices for success. *J Am Med Inform Assoc*. 2014;21(3):464–72.
63. Wright A, Aaron S, Sittig DF. Testing electronic health records in the "production" environment: an essential step in the journey to a safe and effective health care system. *J Am Med Inform Assoc*. 2016;24(1):188–92.
64. Wright A, Sittig DF. A framework and model for evaluating clinical decision support architectures. *J Biomed Inform*. 2008;41(6):982–90.

65. Wright A, Sittig DF. SANDS: a service-oriented architecture for clinical decision support in a National Health Information Network. *J Biomed Inform*. 2008;41(6):962–81.
66. Wright A, Sittig DF. A four-phase model of the evolution of clinical decision support architectures. *Int J Med Inform*. 2008;77(10):641–49.
67. Nadkarni PM, Miller RA. Service-oriented architecture in medical software: promises and perils. *J Am Med Inform Assoc*. 2007;14(2):244–46.
68. Kawamoto K, Lobach DF, Willard HF, Ginsburg GS. A national clinical decision support infrastructure to enable the widespread and consistent practice of genomic and personalized medicine. BMC Med Inform Decis Mak. 2009;9(1):1.
69. Kawamoto K, Lobach DF. Proposal for fulfilling strategic objectives of the US roadmap for national action on decision support through a service-oriented architecture leveraging HL7 services. *J Am Med Inform Assoc*. 2007;14(2):146–55.
70. Kawamoto K, Jacobs J, Welch BM, *et al*. Clinical information system services and capabilities desired for scalable, standards-based, service-oriented decision support: consensus assessment of the Health Level 7 clinical decision support Work Group. *AMIA Annu Symp Proc*. 2012;2012:446–55.
71. Dixon BE, Simonaitis L, Perkins SM, Wright A, Middleton B. Measuring agreement between decision support reminders: the cloud vs. *the local expert*. BMC Med Inform Decis Mak. 2014;14(1):1.
72. Wright A, Sittig DF, Ash JS, *et al*. Lessons learned from implementing service-oriented clinical decision support at four sites: a qualitative study. *Int J Med Inform*. 2015;84(11):901–11.
73. Paterno MD, Goldberg HS, Simonaitis L, *et al*. Using a service oriented architecture approach to clinical decision support: performance results from two CDS Consortium demonstrations. *AMIA Annu Symp Proc*. 2012;2012:690.
74. Paterno M, Schaeffer M, Van Putten C, Wright A, Chen E, Goldberg H. Challenges in creating an enterprise clinical rules service. *AMIA Annu Symp Proc*. 2007;2007:1086.
75. Goldberg HS, Paterno MD, Rocha BH, *et al*. A highly scalable, interoperable clinical decision support service. *J Am Med Inform Assoc*. 2014;21(e1):e55–e62.
76. Dixon BE, Paterno MD, Simonaitis L, *et al*. Demonstrating cloud-based clinical decision support at scale: the clinical decision support consortium. *MedInfo*. 2013;2013:1268.
77. Ash JS, Sittig DF, McMullen CK, *et al*. Multiple perspectives on clinical decision support: a qualitative study of fifteen clinical and vendor organizations. *BMC Med Inform Decis Mak*. 2015;15(1):1.