Healthcare and Medical Devices, Vol. 51, 2022, 5–14
https://doi.org/10.54941/ahfe1002092

Describing and Disarming Health Information System Snares That Capture and Conceal Characters

Tim Arnold[1,3], Helen J. A. Fuller[1], and Angela L. Laurio[2]

[1]Human Factors Engineering, Clinical Informatics and Data Management Office, Office of Health Informatics, Veterans Health Administration, Department of Veterans Affairs, Washington, DC, USA
[2]Informatics Patient Safety, Clinical Informatics and Data Management Office, Office of Health Informatics, Veterans Health Administration, Department of Veterans Affairs, Washington, DC, USA
[3]University of Michigan, College of Pharmacy, Ann Arbor, MI, USA

ABSTRACT

System constraints on computing characters can lead to latent design issues that may contribute to incomplete information and disruptions in workflow. In health information systems, these design issues can lead to unsafe conditions. Currently, prospective design and mitigation efforts are focused on messaging and reporting. Report narratives suggest that there is benefit in this approach but that more rigorous error-tolerant design is necessary to support safer operations. In this paper, we describe the pitfalls realized through the system constraints on computing characters and conceptualize a safety architecture for furthering system error tolerance.

Keywords: Characters in computing, Human factors, Error-tolerant design

INTRODUCTION

In health information entry and exchange, correct encoding and decoding of characters in computing are essential for accurate documentation and interpretation of information. When something goes wrong and there are changes to or deletions of the intended characters, there is missing or inaccurate data in the health information system. Due to the very nature of health information systems, it is difficult to detect when information is missing. Another attribute that is often invisible to users is the way characters are interpreted by computing systems. Lost and invisible information in health information systems can lead to patient safety issues.

Characters in computing are symbols that stand for a single unit of data such as a number, alphabetical letter, or punctuation mark. The Food and Drug Administration (FDA) Maude database (MAUDE, 2022) and the Institute for Safe Medication Practice (ISMP, 2014) describe a few isolated issues with computing characters and health information. The loss of health information due to the failure of correctly translating computing characters could be a problem of potentially high severity. We were unable to locate a descriptive summary of or a plan for fail-safe and error-tolerant systems design for addressing this problem space. The human factors and human-centered design (HCD) communities have the knowledge and skills necessary to address issues related to system states and error-tolerant design. These professionals are equipped to evaluate and offer design solutions to these issues.

In this paper, we review and group issue reports on characters in computing and information entry and exchange. We reflect on human factors and safety engineering principles for designing systems to prevent, detect, and mitigate latent issues in this problem space. Furthermore, we explore special characters that present added challenges when used in computing systems.

To facilitate fail-safe interoperability and health information exchange, systems require designs that address latent issues brought on by hidden attributes of characters and count constraints in computing. Using human factors and safety engineering principles, we can help prospectively design to detect and disarm the snares found within and across health information systems.

APPROACH AND ANALYSIS

During an operational request for information about an isolated incident, we identified a collection of reports describing context around computing characters that lead to unreliable and undesirable system performance. To address these issues, we sought to analyze and concisely describe this collection and use this information to formulate an error-tolerant design strategy.

We used the following search terms to query the collection of reports: character, spec char, hyphen, back slash, and forward slash. Due to time and resource constraints, we were unable to perform an exhaustive search. We retrieved a total of 161 reports. After removing duplicates, we relied on 87 reports for our analysis. Two independent reviewers assessed the reports for applicability and excluded 25 as being primarily related to technical systems issues. As a result, there were 62 reports for further analysis.

We then performed a simple thematic analysis to classify information into computing character issue categories within contexts of work. We sought to understand these issues within the context of work for structuring the problem space and then overlay error-tolerant system design considerations on this framework.

FINDINGS

First, we discuss computing character characteristics and how they may lead to system performance states that may not meet human expectations (Rasmussen, 1983). Then we provide examples of the clinical context in which the system performance expectations were not met. To do this we describe the processes, environments, and people impacted when work systems do not live up to expectations.

Collecting and analyzing computing character limitations and capabilities within contexts of work systems provides the framing for formulating error-tolerant design considerations. Characters in computing are the single symbols used to encode information and can be classified as alphanumeric, punctuation, control, or whitespace characters. These characters are elemental for communicating within and across healthcare systems. However, all system features also have limitations. In this paper, we seek to describe challenges to the design of health information systems in how characters are treated, encoded, and exchanged.

To do this, we arranged the character issues into four groups: character count constraints, special characters, data type mismatch, and miscellaneous character issues. We formed these groupings as an initial step in describing a framework to support error-tolerant systems design considerations.

There were 27 reports that described issues with character count constraints, three describing issues with data types, 22 describing issues with special characters, and 10 reports describing miscellaneous issues with computing characters. The next sections describe in greater detail specifics and examples of each category.

Character Count Constraints

Character count constraints are requirements placed on the number of characters allowed for entry into a system. These limits, which may be a maximum or minimum, are assigned to fields during database design to control the amount of data that is stored. Historically, text strings were limited to 255 characters, because one byte was reserved for string length and because this was a limitation of 16-byte computing (Chen, 2016). Modern database systems can accommodate longer strings (BMC, 2009), and with new field types like BLOB, inherent constraints have all but been eliminated (Oracle, 2012). Twenty-seven reports involved issues with character count constraints. Twenty-five of these issues involved text fields, and two involved numeric fields. The most common results of these limitations were incomplete information being conveyed through the system (10 reports), interrupted workflow (10 reports), and the wrong or no information being displayed (five reports). In one report, character limits interfered with syntactic standardization efforts, forcing users to enter non-standard abbreviations or shorthand as a workaround. Three reports involved conflicting character constraints between two systems, resulting in incomplete information display (two reports) and complete record retrieval failure (one report). The last two issues in this category involved fields designed to store numbers. In one case a field designed to store a decimal number had no constraints on the number of decimal places in one system, while the other system was limited to two places, and the user's workflow and reporting tools supported the continued limit of two. The final report involved a field that allowed for a two- or three-digit integer where a minimum of three digits was required. Displaying one or two digits without preceding zeroes not only violated accepted standards for the field but had the potential to change the interpretation of what was being represented.

Data Type Limits

Data type is an assigned feature of the data. Fields may be designated as containing letters only, numbers only (with sub-categories such as whole numbers, decimals, negative numbers, etc.), a combination of letters and numbers, dates, or dates with times. Like character constraints, data type is determined during the design phase. Two of the reports analyzed involved a field limited to numeric data where alphanumeric was required to meet user needs. The third report in this category accepted alphanumeric characters where only numeric was the requirement. While this last issue was present in a test environment, the fact that the field was used as a patient identifier has serious implications if present in a production environment, especially because the issue was detected in the context of health information exchange, with implications for patient index management. The other two issues occurred as users were documenting patient care. The result in all three of these cases was incomplete data being stored or displayed.

Special Characters

According to the National Institute of Standards and Technology (NIST) Computer Security Resource Center Glossary, a special character is a non-alphanumeric character on a standard English keyboard. It may be assigned special meaning or used to trigger a function, which can vary depending on the application. The standard list of 7-bit ASCII special characters includes those found above the numbers on a standard keyboard plus punctuation marks and mathematical symbols (NIST, 2015). Special characters carry with them different layers of uncertainty including use in different contexts within and across systems (Weinberg, 1971). Special characters may encode for a glyph or have function as a metacharacter depending on the context and system of use. Because special characters may be interpreted differently depending on their context of use, it may be more difficult to predict, detect, and identify unexpected systems states. Because meaning and function can vary among applications, it is not surprising that the use of these characters is associated with reported issues. All the reports in this section involved issues experienced during information exchange, either between different systems or components within one system. Twelve of the 22 reports involved financial components of the system (coding, billing, and revenue cycle). There was no single special character that caused issues, although hyphens and quotation marks were each cited in multiple reports. Special character issues most often resulted in users not being able to access records (nine reports) and data transmission failures (seven reports).

Miscellaneous Issues

There were ten reports that did not fit into any of the three above categories. Six reports involved adding or deleting characters in a field, which resulted in missing data when displayed and mismatched data when transmitted to another system. Six reported errors occurred during information exchange and two while attempting to extract data for routine reporting. In one case an extra character was added to a field used to identify health care providers across systems, resulting in the potential for the wrong provider to be assigned to a patient, order, procedure, or notification. In another case the lack of constraints on the number of characters resulted in text from one field overwriting text from another field when labels were printed, resulting in critical patient identifiers being obscured.

DISCUSSION

The findings described above point to the need to adopt an HCD process when designing systems. By examining the reported issues, it is clear there are human factors considerations related to the concerns and areas where usability principles could be employed for better design.

Human-Centered Design

When designing information systems and systems of systems, at times the requirements specification may be overly focused on the technology and programming requirements and do an incomplete job of considering the needs and capabilities of the human user as well as the relevant environment and perceived constraints. It is important to understand the user's needs both to support the work of the user and to identify and address areas where the user is likely to develop workarounds. HCD should be considered early in the system development process, not only towards the end when user interfaces and dialogs are being refined.

Some reported issues we reviewed suggested the system had not been designed based on a full understanding of the user's work and information system entry needs. For example, one form only supported the entry of numeric characters for lot numbers, but lot numbers included letters so the system should have allowed alphanumeric characters. Another form only accepted single or hyphenated last names, whereas double last names are common in the cultures of some system users. Other reports described data fields that were not large enough to support intended entries. A multipronged approach to address these issues should include thorough consideration and documentation of work requirements and information system entry needs earlier in the system design and might consider initial design iterations with larger fields and flexible data types followed by analysis to inform decision-making about database architecture. Design teams should consider decisions in conjunction with user messaging and training.

It is very important to understand Work as Done (WAD) as opposed to Work as Imagined (WAI), because what users are actually doing yields important information about work requirements, environmental constraints, and user preferences (Braithwaite et al., 2017; Conklin, 2012). If you fail to understand how users are using systems, any subsequent design or training efforts will be based on an incorrect or at least incomplete understanding of the current state. This speaks to the importance of thoughtful planning of change management activities to decrease errors and support users in learning a new system. Considerations should include matching the user's expectations—including those based on the functionality of previous systems—and informing users of changes. Only by understanding WAD and user goals and requirements can designers create systems and interfaces that meet user needs.

In addition to having systems that perform as users desire and expect, it is important to promote timely and relevant feedback to the users to keep them informed of the system state. This might include conducting systems checks and providing feedback when a user entry is expected to lead to an error or blocking that entry as well as giving point-of-use instructions on what entry characteristics are—or are not—permitted. Supporting recognition rather than recall, visibility of system status, and speaking the user's language are common heuristics for HCD that support usability and error-tolerant design (Nielsen, 1994).

Error-Tolerant Design

As described in the previous section, the design of system components should facilitate putting knowledge in the world and providing visibility of system status through human-centered features while speaking to users in a language they understand. When designed with these principles in mind, the system also is designed to prevent issues by fostering a dialog, bringing awareness to current and potential states while drawing attention to unexpected performance and when needed assisting users in recovering from undesired pathways in a helpful and efficient manner.

To do this, we conceptualize communication design in and across the work systems as a continuum of conversations where humans and computers may enter, engage in, contribute, and leave throughout ongoing health care services. When socio-technical systems linguistically align and are designed to be helpful, there is a reciprocity effect (Branigan et al., 2010). In health information systems, it is likely that if linguistic alignment is to occur it is through a communicative process of ongoing dialog between humans and computers. This alignment may happen through manual design of interfaces, but moving forward this may become more and more automated (Kraska et al., 2019). As a result, it is important to formulate effective and flexible safety design patterns that can carry us into increasingly automated systems.

Designing for linguistic alignment and continuous dialog in facilitating error-tolerant features should be a primary focus in planning and managing safety-critical systems. Wood & Kieras (2002) describe an error-tolerant system as a system that is designed to prevent, identify, detect, correct, mitigate, and monitor for errors and latent design issues. To guard against unexpected system performance, it is often necessary to have multiple touchpoints and feedback throughout the work tasks. The design of touchpoints and dialog throughout can inform preventative design and help bring awareness to unexpected performance and concerning future states.

Prospective strategies such as preventive design approaches and messaging can help avoid undesired system performance. Preventive approaches include designing the interface to provide visibility of critical system capabilities and limitations. For example, when a maximum character limit is deemed necessary and assigned, the interface should be designed to inform the user in real time of the character limits (U.S. Web Design Systems, n.d.). A character counter that visibly displays a running count during entry may assist the user in recognizing the availability of space left. Human

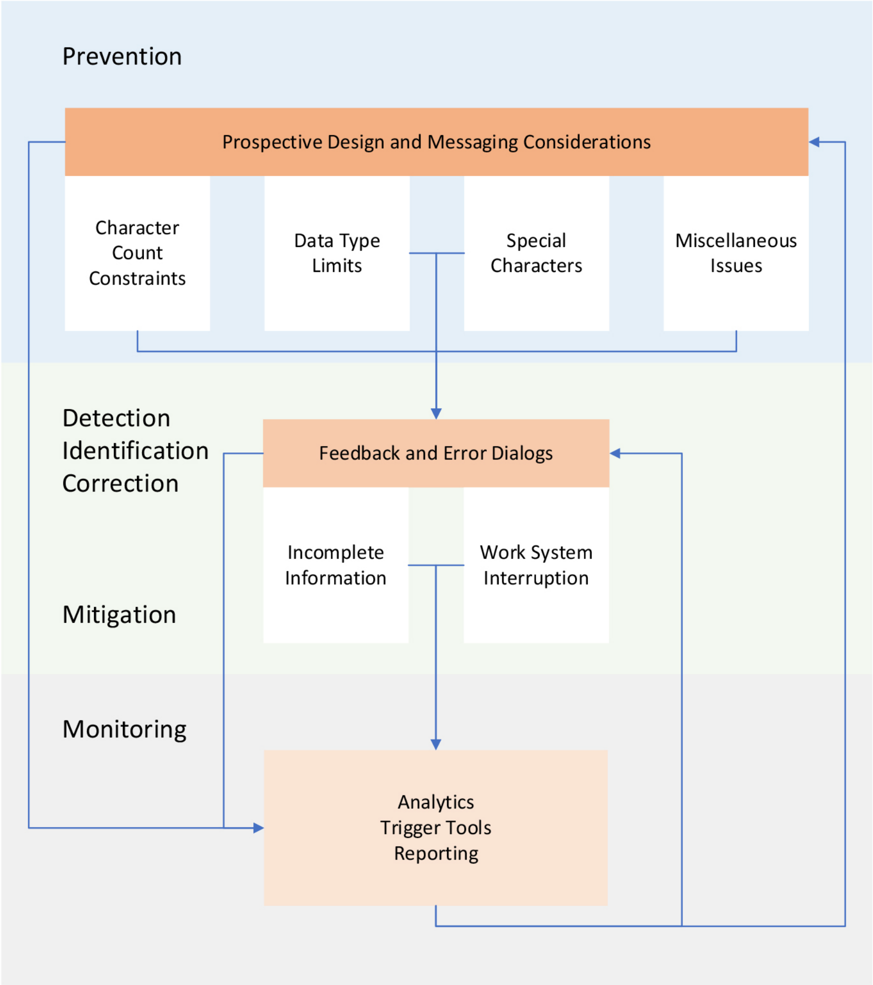

Figure 1: System design considerations for preventing, detecting, and disarming latent issues associated with computing characters in health information systems.

factors and HCD professionals can help database designers understand users' needs and weigh tradeoffs when making decisions about which data types to use.

When the interface allows user entry of characters outside systems limitations, the system should alert the user in words that they understand. Messaging should provide the user control in reducing unexpected system performance (Figure 1). The message should provide useful and usable feedback about next steps and should provide closure that the issue has been or will be resolved (Schneiderman et al., 2017). Default text within some reports suggests that reporters may be prompted to copy error messages into the reported issue. A few reports in this analysis describe the error message language used to communicate issues. Messaging encouraging users to enter the error message they witness is a form of *repeat back* that facilitates dialog within the system. Adding a process for actively monitoring error messages and responses and iteratively evaluating the usability of messaging are options for understanding the impact of error messages within systems. The next section expands on surveillance as a strategy for cultivating awareness and communication.

In Figure 1, monitoring, surveillance, and reporting are shown connecting to other error-tolerant design considerations in a network of continuous dialog, bringing light to systems states for interlocutors working in the system. Monitoring user input for special characters, character limit issues, and challenges to existing data types in the system of entry and mismatches in data exchange systems can help design teams make decisions about data field types and sizes that meet users' needs and reduce unexpected performance. Error message handling reviews and monitoring can help to understand the problem space and prioritize improvement activities. During creation and redesign, error messages should be co-produced, usability testing performed, and documentation provided in the user help section (Schneiderman et al., 2017). Voluntary reporting only captures some issues but is important to an overall surveillance program and can inform future prospective design efforts.

Surveillance and monitoring efforts should not only capture concerns and latent design issues but should also consider positive variation or context-based ways of doing work that are beneficial and error-tolerant. Although the design considerations described here are seemingly intuitive and based on design principles, they should be considered in the larger context of systems design. Thoughtful consideration should be given to design trade-offs and their potential influence on the greater system. Systems should be examined, evaluated, and tested to further understand overall performance over the range of uncertainties.

Limitations

A limitation to this analysis is that reports often lacked information regarding where and how these issues happened. Reporting will typically not capture all the necessary information to perform rigorous systems studies (Jha et al., 1998; Meyer-Massetti et al., 2011). Additional discussion with the reporters and work system operators may be beneficial and necessary to expand understanding. In future discussions, we will describe the analysis of similar reporting systems and draw from those to fill in some of the gaps in where and how these issues take place. This work may also highlight the value in having multiple reporting systems with slightly different structural elements in facilitating the collection of informational variants.

CONCLUSION

In this paper, healthcare work information systems are discussed, but these concepts could be used to think about and inform patient-facing information systems. Furthering a communication continuum focused on bringing awareness and useful next steps into patient-facing applications should be designed to accommodate a variety of users and communities (Holden et al., 2020).

We recognize the importance of focusing design on flexible dialog in socio-technical systems. It is critical in healthcare that technology is designed to facilitate visibility of system states through periodic messaging and cueing while bringing awareness to the range of uncertainties (De Neufville and Scholtes, 2011) and unexpected performance capturing the essence of effective communication amongst teammates.

ACKNOWLEDGMENT

We thank those working at the Veterans Health Administration, Department of Veterans Affairs for their commitment to designing safer systems. There were no relevant financial relationships or any source of support in the forms of grants, equipment, or drugs. The authors declare no conflict of interest. The opinions expressed in this article are those of the authors and do not necessarily represent those of the Department of Veterans Affairs.

REFERENCES

Braithwaite J, Wears RL, Hollnagel E, editors. (2017). Resilient Health Care. Vol. 3. Reconciling work-as-imagined and work-as-done. Boca Raton (FL): CRC Press, Taylor & Francis Group.

Branigan, H. P., Pickering, M. J., Pearson, J., & McLean, J. F. (2010). Linguistic alignment between people and computers. Journal of pragmatics, 42(9), 2355–2368.

Chen, R. (December 29, 2016). The evolution of the text size limits related to the standard static control. The Old New Thing (blog). Available at: https://devblogs.microsoft.com/oldnewthing/20161229-00/?p=95045

Conklin, T. (2012). Pre-accident investigations: An introduction to organizational safety. Ashgate Publishing, Ltd.

De Neufville, R., & Scholtes, S. (2011). Flexibility in engineering design. MIT Press.

Holden, R. J., Cornet, V. P., & Valdez, R. S. (2020). Patient ergonomics: 10-year mapping review of patient-centered human factors. Applied ergonomics, 82, 102972.

Institute for Safe Medication Practice (ISMP). (2014). Potential inaccuracy of electronically transmitted medication history information used for medication reconciliation. Available at: https://www.ismp.org/alerts/potential-inaccuracy-electronically-transmitted-medication-history-information-used

Jha, A. K., Kuperman, G. J., Teich, J. M., Leape, L., Shea, B., Rittenberg, E., ... & Bates, D. W. (1998). Identifying adverse drug events: development of a computer-based monitor and comparison with chart review and stimulated voluntary report. Journal of the American Medical Informatics Association, 5(3), 305–314.

Kraska, T., Alizadeh, M., Beutel, A., Chi, H., Kristo, A., Leclerc, G., ... & Nathan, V. (2019, January). Sagedb: A learned database system. In CIDR.

MAUDE-Manufacturer and User Facility Device Experience. (2022). Available at: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfmaude/search.cfm.

Meyer-Massetti, C., Cheng, C. M., Schwappach, D. L., Paulsen, L., Ide, B., Meier, C. R., & Guglielmo, B. J. (2011). Systematic review of medication safety assessment methods. American Journal of Health-System Pharmacy, 68(3), 227–240.

Nielsen, J. (1994, April). Enhancing the explanatory power of usability heuristics. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 152–158).

NIST, Information Technology Laboratory. (2015). Special character - glossary. CSRC, National Institute of Standards and Technology. Available at: https://csrc.nist.gov/glossary/term/special_character

Oracle. (2012). BLOB data type. Getting started with Apache Derby. Oracle Inc. Available at https://docs.oracle.com/javadb/10.8.3.0/ref/rrefblob.html

Purdue University. (2009). Field limit specifications: BMC Software. Available at: https://support.purdue.edu/help/FootPrintsHelp/content/field_limit_specifications.htm

Randhawa, G. K., Garnett, A., Huang, S., Dhot, P., & Fyfe, M. L. (2019). Evidence-based usability principles for safe computerized provider order entry (CPOE) Interface Design. MEDINFO 2019: Health and Wellbeing e-Networks for All, 1947–1948.

Rasmussen, J. (1983, January). Design for error tolerance. In Proc. Winter Meeting of Amer. Nuclear Soc.

Schwartz, B., Zaitsev, P., & Tkachenko, V. (2012). High performance MySQL: optimization, backups, and replication. " O'Reilly Media, Inc."

Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., & Elmqvist, N. (2017). Designing the user interface: strategies for effective human-computer interaction. Pearson.

U.S. Web Design System (USWDS). (n.d.). Character count. Available at: https://designsystem.digital.gov/components/character-count/

Weinberg, G. M. (1971). The psychology of computer programming. p. 221. New York: Van Nostrand Reinhold.

Wilkerson, M. L., Henricks, W. H., Castellani, W. J., Whitsitt, M. S., & Sinard, J. H. (2015). Management of laboratory data and information exchange in the electronic health record. Archives of Pathology and Laboratory Medicine, 139(3), 319–327.

Wood, S. D., & Kieras, D. E. (2002, December). Modeling human error for experimentation, training, and error-tolerant design. In Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (pp. 1075–1085).