

Investigating Delays in Updates to Infusion Pump Drug Limit Libraries

Poching C. DeLaurentis, Ph.D.¹, Kang-Yu Hsu, M.S.^{1,2}, Ana Isabel De la Hoz Armenta^{1,3}, Yuval Bitan, Ph.D.⁴

¹Regenstrief Center for Healthcare Engineering, Purdue University, West Lafayette, IN;

²School of Industrial Engineering, Purdue University, West Lafayette, IN; ³Universidad Nacional de Colombia, Medellín, Colombia; ⁴Department of Industrial Engineering and Management, Ben Gurion University of the Negev, Be'er Sheva, Israel

Abstract

Interoperability is a major challenge in current healthcare systems. It brings big hope for data exchange, but also raises some concern about patient safety. We study the wireless updating of modern infusion pumps and demonstrate the possible flaws in this process. Through analyzing data on drug limit libraries (DLL) versions in one hospital we could identify the delays in distributing DLL updates and the impact these delays might have on patient safety. We found that 31% of all started infusions had used outdated DLL versions, and 22.6% of all alerts were triggered by outdated DLLs. These findings suggest that clinical and operational stakeholders in healthcare systems must address the unreliable interoperability of medical technologies such as seen on infusion pumps. The impact of information inconsistency across healthcare systems might result in use error which would impair patient safety.

Introduction

Medical devices' advancement in automation and interoperability is playing a significant role in the way they are integrated into the drug delivery system. In order to make all component synchronized in the integrated system, which includes the human operator and the medical devices, it is important to have very reliable communications between all of them. Devices that are not synchronized may cause inconsistency (or, from the operator's viewpoint, operational failure) which in turn might translate into use errors and distrust of the technology. Such distrust likely would lead to low adoption rate of using the technology and also undesired workarounds^{1,2}.

Infusion pumps are among the technologies that have advanced interoperability in the clinical setting. Modern infusion pumps models have built-in dose error reduction systems (DERS) and drug limit libraries (DLL) that can alert users when a drug is programmed outside of the preset limits, and they have shown to be effective in reducing potential infusion errors³. These functions were designed as a safety feature that should ease and standardize the way clinicians program the infusion pump. Moreover, hospitals often make changes to drug limits to reflect evolving clinical needs, and, thus, would publish new versions of DLL from time to time. Such updates on DLL are done wirelessly nowadays without the need to perform updates of the DLL file on each individual pump manually with a cable. Ideally, in order to standardize clinical workflow and avoid confusion, all infusion pumps in the hospital should have the same version of the DLL especially with such wireless automation.

Objective

The objective of this study is to raise clinicians' awareness of the issue of inefficient infusion pump DLL updates, and what the impact may be. The problem of delayed updates on pumps may have been known by some but not so much about its magnitude, its impact on drug limit compliance and the potential risk of patient harm^{4,5}. Following the authors' prior descriptive work on the issue of long update delays of DLL in several hospital systems using merely infusion alert data⁵, we took a step further to collect pump status and all infusion detail reports in addition to the alert data from one of our collaborating hospitals. By cross checking the numbers in each of these different types of report, we would also be able to discover in-depth the spread and impact of such issue on infusion drug alerts, potential patient harm and its contribution to nursing workarounds.

Materials and Methods

Data

The data for this analysis came from three infusion pump data sources. One is drug alert data from the Infusion Pump Informatics (IPI) System maintained and supported by the Regenstrief Center for Healthcare Engineering (RCHE) at Purdue University⁶. Another is the pump status report which can be generated by a vendor pump management software. The other data source is the detailed all infusion report, available through the vendor analytics software tool,

which contains each infusion start of the dates of interest, with or without triggering drug alerts or alarms. Each report type is explained as follows.

Infusion Pump Alert Data. All IPI community member hospitals share pump alert data by periodically uploading them to the IPI System on Catalyzecare, a Hubzero[®]-based cyber infrastructure. An infusion alert happens when a programmed value violates the preset limits for the drug and its clinical use, and the clinician can decide to accept the alert by cancelling or reprogramming the infusion or override the alert. As of late 2015, the system contained more than 21 million infusion pump drug alerts from more than 120 hospitals across the U.S. Even though the IPI data contains alert data from multiple pump manufacturers, we focused our analyses using data from one of them. Each line of alert includes detailed information such as drug name, profile name, drug limit library version used, infusion type, programmed value, drug limit violated, time stamp of infusion attempt, etc.

Pump Status Reports. Pump status reports are available through the vendor provided pump management software tool which is different from the analytics suite. A pump status report is a snapshot of all the pumps in the network. For each pump it details the location (facility), model, pump serial number, activated DLL (i.e., the version all pumps should be using at the moment), DLL version on the very pump, any pending DLL version on the pump (i.e., downloaded DLL file but yet to be installed), last communication time stamp (between the pump and the server) and profile last observed.

All Infusion Detail Reports. As the name suggested, all infusion relevant events are included in an all infusion detail report, such as infusion started, restarted, stopped, paused, alerted, resolved, time stamp, whether or not the event is under guardrails system, type of infusion (continuous, bolus, etc.), actual infusion duration seconds, programmed duration seconds, volume to be infused, dosage, and infusion rate, etc. An infusion delivery is actually composed of a sequence of infusion events which share the same infusion ID. As show in the report, usually an infusion administration starts with an “infusion started” event and ends with an “infusion stopped”. In summary, the all infusion report lists information regarding when an infusion event happened, for what purposes it happened, how the infusion was programmed (using basic mode or with guardrails), and the detailed infusion settings. This allows us to study not only the infusion alerts generated by various DLL versions but also the bigger picture of all infusion starts that did not generated any alert.

Methods

We used the alert data of an inner-city hospital of 315 beds in the Midwest from July 1st to September 18th, 2015 and the DLL version information indicated in each alert to show how many and which DLL versions exist in the infusion drug alert. There were 680+ pumps (i.e., the main units with memory) in the facility. We also collected the hospital’s pump status reports and all infusion detail reports for the same period of time. Since the pump status report is a snapshot of each pump’s status in the fleet, we requested that the pharmacist in charge to send one to us daily if at all possible in order for us to keep track of the status change of each pump and the patterns of the changes. Our partner hospital had three DLL updates during that period of time. Note that an update cycle is defined as the time from the first day of a new published version of DLL to the day before the publication of the next DLL version. We then classified, on that specific reporting day, each pump as “current” or “outdated” by comparing the DLL version on each pump with the activated DLL version. We further examined the severity of the DLL update problem in all infusions recorded in the all infusion detail reports.

Results and Discussion

DLL Update Rate by Alerts

Figure 1 illustrates the phenomenon that alerts could be generated by outdated DLL versions. Each color represents the alerts trigger by a DLL version, and each column is a day. In this hospital, alerts were seen to be generated by seven different DLL versions within an 80-day window. As the figure shows, once a DLL update is pushed out to the pumps, alerts by the new version started to happen (as a new color starts to show in the bars). However, the more troubling is the fact that some alerts were still being triggered by updated DLL versions as their colors bleed into more recent days. Should the drug limit settings be different in the new versus the old DLL versions, clinicians may be getting false alerts (i.e., those that should not have happened) or missing alerts (i.e., those that should have happened).

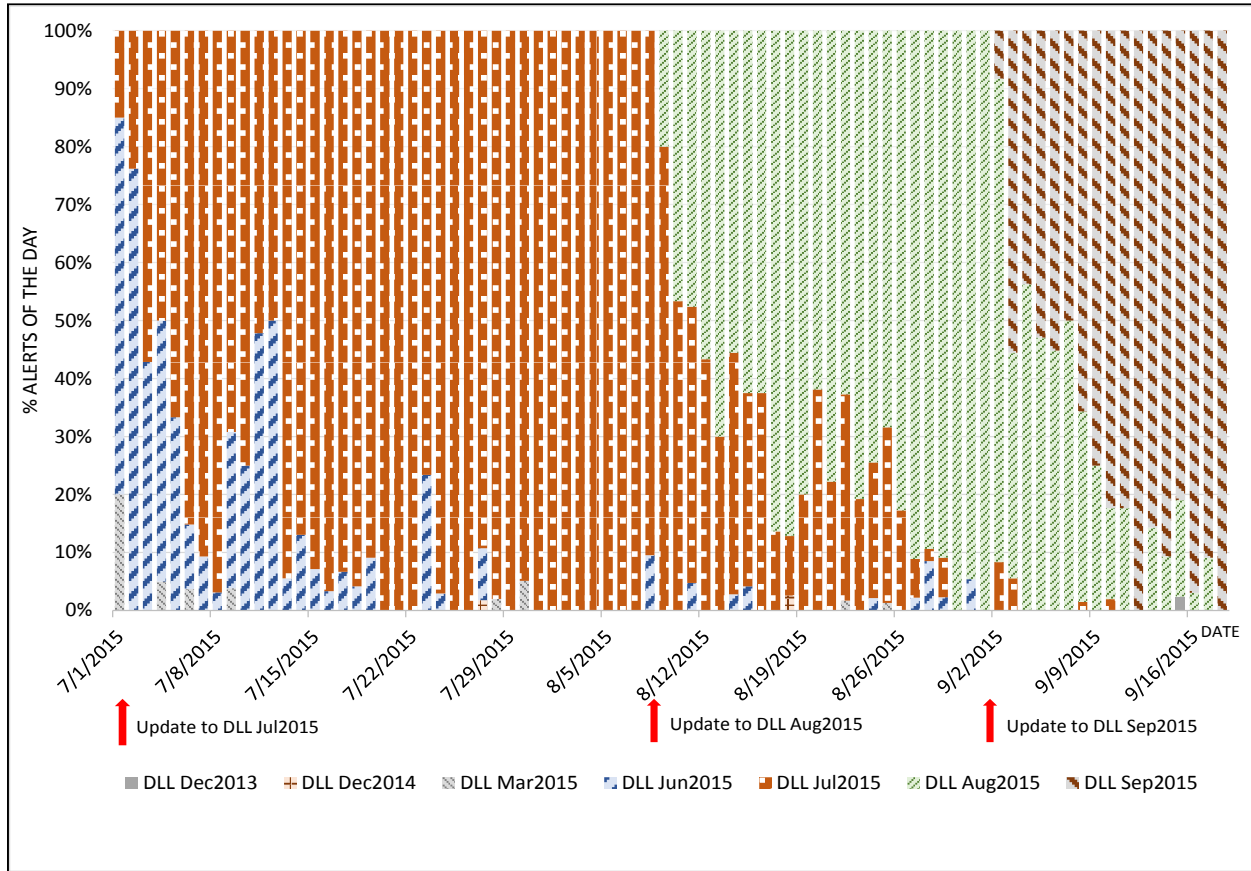


Figure 1. Percentage of alerts by DLL versions.

DLL Update Rate by Pump Status Reports

We tracked each individual pumps and their status using the pump status reports. Figure 2 shows the distribution of the DLL versions on each day of the reporting. Each time a new color appears in the vertical bars it indicates that a new DLL version has been released and has been installed on some of the pumps. However, as shown in the figure, more and more colors show up in the bars as time goes. It means that the number of DLL versions kept increasing every time a new one is released, i.e., the old versions never got replaced on some pumps during this period of time.

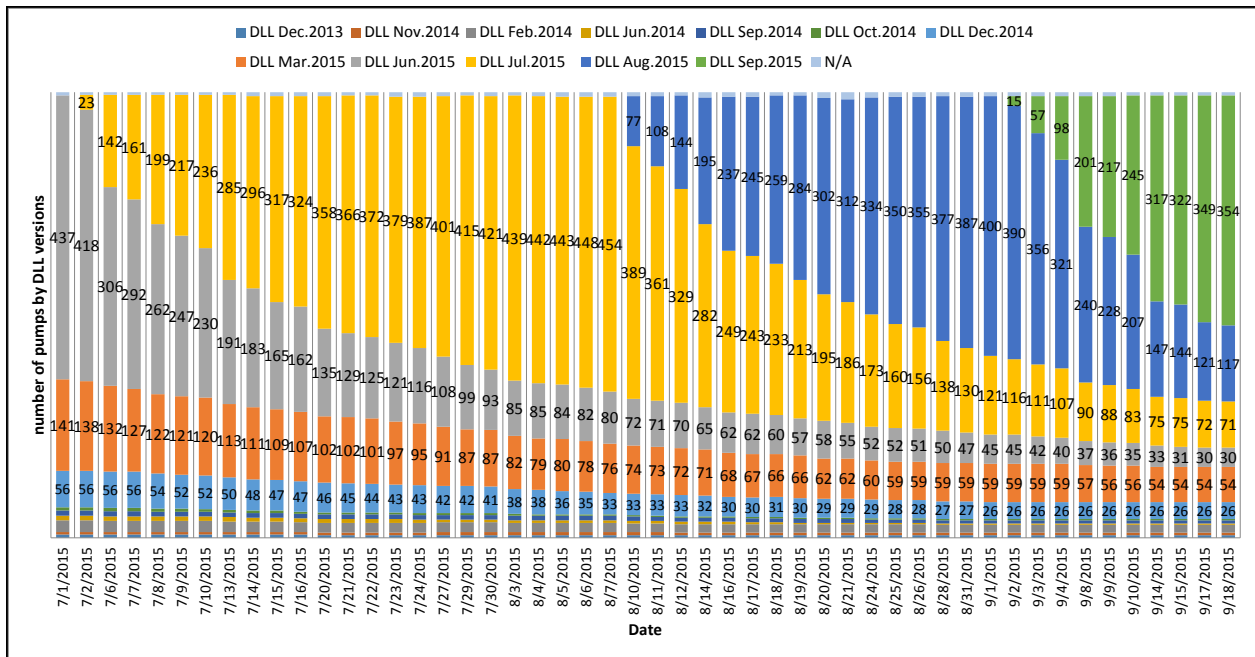


Figure 2. The number of pumps (shown) of each DLL version. The length of each color on the bar indicates the percentage of such version found on that day.

We further showed the effect of the pumps using outdated DLL versions by combining the alert counts and the status of each of the pumps on these days. Figure 3 shows the correlation of them: the diamond markers are percentages of pumps with outdated DLL based on pump status reports, and the square markers are the percentages of alerts generated by outdated DLLs. Naturally, the more pumps with outdated DLL were published and pushed out to all pumps in the network. If the probability of alert generated per pump and the utilization of pumps be equal on all, these two percentages should remain very close to each other on any given day. One possible explanation of the gaps between the diamond and square markers on some days may be that some pumps were not being used as often as others were. In other words, it is reasonable to speculate that some pumps were used and/or turned over much more often thus they were provided more opportunities to get updated and be generating alerts. More analysis is needed for us to understand this phenomenon better.

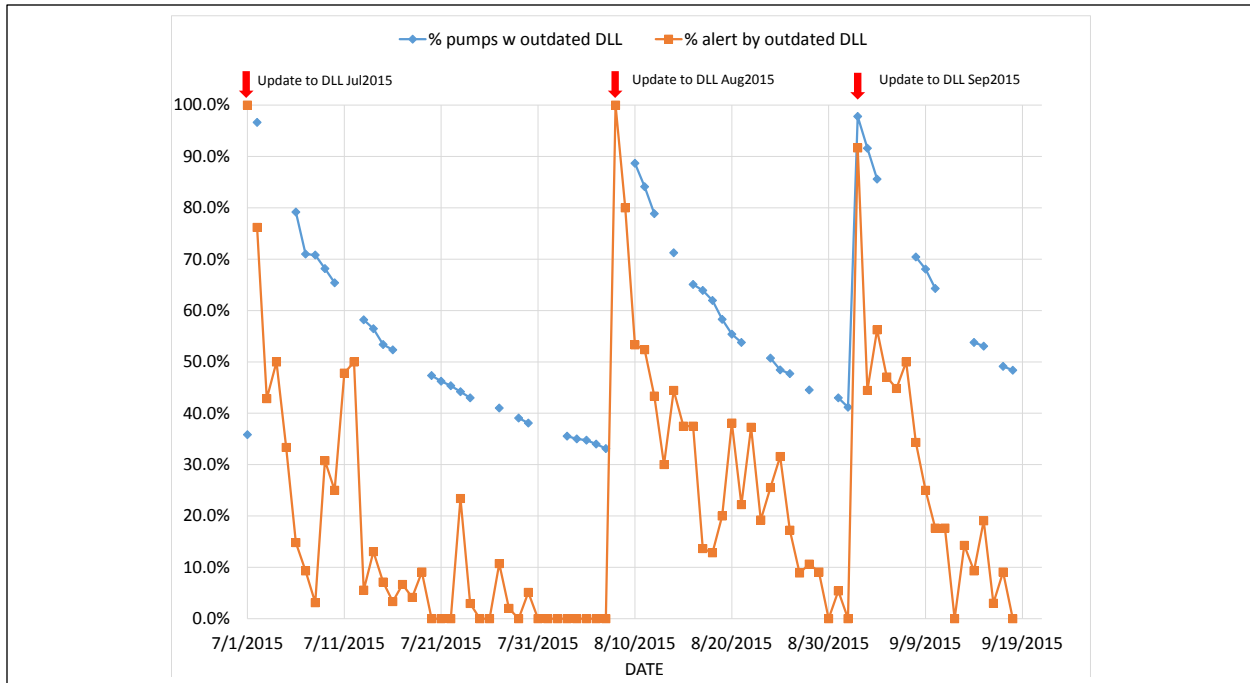


Figure 3. Percentages of pumps and alerts by outdated DLL versions.

Impact of Using an Outdate DLL Version on the Pump

We extracted all events of infusion started from the all infusion detail reports during this period of time and investigated how many of started infusions were on pumps with outdated DLL versions. The results were further categorized by infusion type (Table 1). It shows that on average almost one third of all infusions started with an outdated DLL version, and, among all infusion types, bolus infusions had the largest incident rate (44%). The results here also indicate that over 30% of all attempts of infusion starts were at risk of not equipped with the most up-to-date version of the DLL on the pump. Recall the DLL analysis on infusion drug alerts seen earlier in this section, the results of infusion started events showed us that drug alerts do only represent a small portion of all that might have been “risky” infusions.

Table 1. Numbers and percentages of infusion started events performed on pumps with outdated DLL versions from July 1 to September 18, 2015.

Infusion Type	Total # Infusion Started Events	# of Infusion Started Events with Outdated DLL Versions (%)
Continuous	28150	10011 (36%)
Bolus	10432	4548 (44%)
Fluids	31640	8288 (26%)
Intermittent	32507	9147 (28%)
Total Number	102729	31994 (31%)

Conclusion

In this study we exhibited and verified long delays of infusion pump drug limit library updates using multiple data sources from one hospital including pump drug alerts, pump status reports, and all infusion detail reports, all available through vendor provided software tools. In this specific case study of one hospital we saw that on average more than 50% of the pumps were using outdated DLL on reporting days, 31% of all started infusions had used outdated DLL versions and 22.6% of all alerts were triggered by outdated DLLs. This has great implication of potential inconsistency

of pump response to the operator, and the possibility of patient harm due to incorrect drug limit settings (such as when an error made in the DLL was not able to be corrected in time for patient care). Outdated DLL triggered false alerts may cause confusion of the nursing staff and attribute to alert/alarm fatigue problem well known in the healthcare setting while missed alerts may allow inappropriate infusions to be administered. It is also very likely that a nurse would choose to use “basic mode” of infusion, i.e., electing not to apply drug limit settings to the infusion, when s/he cannot find the drug due to not having the most up-to-date DLL on the pump. Such workaround or deviation from the best practice of using the limit settings (i.e., noncompliance), is highly undesirable because of its potential to cause patient harm should a programming error occurs.

Our analyses provided different ways that a hospital can utilize pump informatics to track the effectiveness of pump updates and monitor their progression and impact. We believe it is crucial in ensuring consistency in nursing workflow, thus reducing unnecessarily workarounds, and most importantly, patient safety in hospital care. Our future work will include more in-depth analyses on missed or false alerts due to outdated DLLs.

References

1. Halbesleben JR, Savage GT, Wakefield DS, Wakefield BJ. Rework and workarounds in nurse medication administration process: implications for work processes and patient safety. *Health Care Manage Rev* 2010;35(2):124-33.
2. Tucker AL, Heisler WS, Janisse LD. Designed for workarounds: A qualitative study of the causes of operational failures in hospitals. *Perm J* 2014;18(3):33.
3. Ohashi K, Dalleur O, Dykes PC, Bates DW. Benefits and risks of using smart pumps to reduce medication error rates: a systematic review. *Drug Saf* 2014;37(12):1011-20.
4. Poppe LB, Eckel SF. Evaluating an approach to improving the adoption rate of wireless drug library updates for smart pumps. *Am J Health Syst Pharm* 2011;68(2):170-5
5. DeLaurentis PC, Bitan Y. Can clinicians trust infusion pumps interoperability? A descriptive analysis of the delays in drug limit library updates. Under review 2016.
6. Witz S, Buening NR, Catlin AC, et al. Using Informatics to Improve Medical Device Safety And Systems Thinking. *Biomed Instrum Technol* 2014;48(s2):38-43.