Automated, Web-Based Solution for Bidirectional EHR–Infusion Pump Communication

Steven M. Griffiths, Adam Sapirstein, Julio C. Guzman, Zaza Soriano, and Alan D. Ravitz

Abstract

Smart Agent is a web-based solution for establishing bidirectional communication between an infusion pump and an electronic health record (EHR). It eliminates the need for clinician double check of medication administration using an infusion pump. Because the clinician already is using the EHR to review patient health information and update status, the addition of the web service would help eliminate the potential for human error when using a manual system. The Smart Agent process encompasses the reading of pertinent patient data from the EHR, determination of a new medication dosage based on an internal protocol, input of the dosage into an infusion pump, confirmation of the medication dosage acceptance at the infusion pump, and recording the medication change back into the EHR. The widespread use of Smart Agent–type algorithms with bidirectional communication capabilities would result in safer, more efficient provision of care, as well as better value.

Since the introduction of the medication infusion pump 45 years ago, the healthcare industry has explored ways to save time while reducing and eliminating adverse drug events (ADEs) resulting from human error in the setup and programming of pumps. The introduction of the smart pump, which includes dose-error reduction systems and drug libraries, has reduced but not eliminated ADEs arising from transcription errors, incorrect algorithmic determination and programming of doses by providers, and other typical human-in-the-loop miscues. To make further gains in patient safety and workflow improvements, a broader design of the medication administration process should be considered. This design can leverage emerging capabilities of the electronic health record (EHR) and medication infusion pumps by integrating the best practices of medication administration into an integrated workflow.

The process encompasses the bidirectional transfer of patient data to and from an EHR, as well as the determination, input, and confirmation of the medication infusion dosage to the infusion pump.

Medication infusion pumps are used in a number of administration scenarios, ranging from timed single doses to titrated infusions. The evolution of these devices has ensured that they can be part of a process to follow the five rights of medication administration: (1) right patient, (2) right drug, (3) right dose, (4) right route, and (5) right time.

With the introduction of and reliance on mechanization and computer science, the automation of medication administration continues to develop rapidly. Scientists and engineers in healthcare continue to examine automated control and interoperable plug-and-play systems to connect devices, improve data flow, and save resource time. However, although the community has been discussing medical device interoperability for many years, obstacles to automation remain. Many medical devices share data with EHRs using open-source languages such as Health Level-7 (HL7), but the systems do not have the means to manipulate data, make patient treatment decisions, or update settings on other devices.

To address this technology gap, King et al. demonstrated how a medical device coordination framework could be used for physiological closed-loop control (PCLC) of drug delivery with a patient-controlled analgesia (PCA) pump. In their study, the physiological information obtained by a bedside medical device (i.e., a pulse oximeter) was continually monitored by a “supervisor” system that analyzed the data and disabled the PCA pump if a decrease in the patient’s arterial oxygen saturation indicated respiratory depression. The goal of this type of work was to reduce reliance on manual interventions and improve safety by using near-real-time patient physio-
logical data to directly control devices. The Food and Drug Administration (FDA) is working with the medical industry to regulate and guide the introduction of PCLC systems.11

Another area of interest for PCLC systems is the protocol-guided administration of insulin, which has the potential to become a completely automated process. In the current manual process, the clinician takes a patient blood glucose (BG) reading at the point of care using a hand-held glucometer, then enters the value into the EHR. The clinician reads the patient’s previous BG level from the EHR. Using the patient’s previous and current BG levels and current insulin infusion dose, the clinician calculates the new insulin dose level using the algorithm in the hospital’s insulin protocol. The clinician then sets the new insulin dose in the infusion pump, starts the new insulin rate, and enters the new insulin dose rate into the patient’s EHR. In the case of insulin and other high-risk medications, before the start of the new infusion rate, a second clinician is called to verify that the determination was performed correctly, the new dose was entered into the infusion pump properly, and the new insulin infusion rate was entered appropriately into the patient’s EHR.

This fully automated process, which industry and the FDA are interested in attaining,11 would greatly reduce the amount of clinician time required for each periodic patient check and subsequent medication infusion rate change.

At hospitals throughout the United States and around the world, BG levels are determined and insulin is infused primarily using manual methods. Until the process can be moved closer to automated control, clinicians will continue to experience errors in calculating new doses and in entering dosages correctly into the medication infusion pump and EHR. Valuable clinician time also will be consumed determining dosage levels and double checking other clinicians’ work.

At The Johns Hopkins Hospital (JHH) in Baltimore, MD, the practice has moved only slightly from the fully manual process. The collection of blood and determination of BG levels are done automatically using a network-connected glucometer. For each patient, this BG level is automatically downloaded to the patient’s EHR. The remainder of the process, including pulling required information (BG levels and current dose) from the EHR, determining the new insulin dose rate (using the hospital’s adult insulin algorithm, which is based on the Yale Insulin Infusion Protocol12), programming the insulin pump, and double checking the nurse’s calculations and actions, is carried out manually. Despite an obvious need, no decision support or closed-loop medical device control occurs.

This article describes an example of the current state of the practice, in which limited medical device coordination is available. It also details the work of the Johns Hopkins University (JHU) Applied Physics Laboratory (APL) and Armstrong Institute for Patient Safety and Quality at Johns Hopkins Medical Institute (JHMI) in developing Smart Agent—an algorithm that automates the coordination of medical devices more fully with information made available to the hospital EHR. The authors assert that automated physiological data analysis and dose determination would improve patient safety.13

Technical Approach
As part of an Agency for Healthcare Research and Quality grant,13 scientists and engineers at JHU APL, along with scientists and clinicians at JHH, developed a web-based application called Smart Agent. This application monitors the EHR for patient BG levels on an “as-called” basis and automatically calculates and sends new clinician-selected insulin doses to the infusion pump user interface (UI). From the UI, the clinician can verify the appropriate dose prior to accepting the new dose on the infusion pump. The Smart Agent Web Service (SAWS) automates the process for continuous intravenous (IV) insulin administration considerably. For this project, Smart Agent was deployed to be used with the JHH EHR system (Epic; Epic Systems Corporation, Verona, WI).

Smart Agent System Description
The Smart Agent system is an analysis program incorporated into the hospital’s information system that serves as an interface between a hospital EHR and an insulin infusion pump. At the heart of the analysis is the algorithm that is routinely used for

Until the process can be moved closer to automated control, clinicians will continue to experience errors in calculating new doses and in entering dosages correctly into the medication infusion pump and EHR. Valuable clinician time also will be consumed determining dosage levels and double checking other clinicians’ work.
determining doses of continuous IV insulin. Smart Agent runs on a virtual machine and can be deployed to any network within the hospital system, provided it has access to the hospital EHR and communication with the infusion pump. Within Smart Agent is a series of routines, one of which downloads patient BG values (current measurements and last accepted measurement) and current insulin dose rate from the EHR. The rate that is listed in the EHR is checked against the actual infusion rate on the pump.

The Smart Agent algorithm then takes the BG and dose values and, using the hospital insulin algorithm, calculates the updated insulin dosage for the patient. The clinician must accept this new dose at the Smart Agent UI before the new dosage level is sent through the network to the specific patient-assigned infusion pump. The pump UI is then automatically loaded with the new dosage. The clinician must approve the dose on the infusion pump to initiate the new infusion rate. The infusion rate and selected BG level are then stored in a data sheet within the EHR.

**Architecture**

**Physical infrastructure.** Smart Agent physically interacts with the Epic EHR system at JHH. Specifically, the research is conducted by interacting with the Epic test or development environment at JHH. Although the work was originally conducted through a network bridge between JHH and JHU APL, it was made mobile by moving the SAWS and the MedNet/Mirth virtual machines to a laptop host machine. The Oracle VM VirtualBox on the host laptop then communicates through an unmanaged switch to a Cradlepoint LTE (Long-Term Evolution) router and the ICU Medical (formerly Hospira) Plum 360 medical infusion pump (Figure 1).

The SAWS (Figure 2) serves as the central integration point to connect external entities, such as an EHR, to a medical device. The current system also serves as an interface to calculate insulin infusion rates using the Smart Agent Insulin Infusion Algorithm Java Library table, which was developed by JHU APL using the JHH insulin infusion algorithm to include all possible combinations of the current BG reading ranges, BG value changes, and rate changes for the current insulin dose rate. The communication formats used are JavaScript Object Notation (JSON) and HL7.

The SAWS system consists of six main software components, as shown in Figure 2:

1. Service controller: service that exposes the URL (Universal Resource Locator) interfaces that execute function calls by

---

Figure 1. Physical architecture: diagram of equipment involved in the Smart Agent system. Abbreviation used: APL, Applied Physics Laboratory.
receiving and responding to messages

2. **MongoDB**: embedded database used for storing pump data

3. **HL7Sender**: Transmission Control Protocol (TCP)/Internet Protocol (IP) and Minimum Lower-Layer Protocol (MLLP) software server listening on port 9111 that sends new infusion rate data in a HL7 message format to MedNet/Mirth and replies to HL7 acknowledgment (ACK) messages from MedNet/Mirth.

4. **HL7Receiver**: TCP/IP and MLLP software server listening on port 9112 that receives current infusion pump data containing the current infusion rate in a HL7 message format (from MedNet/Mirth) and replies with HL7 ACK messages (to MedNet/Mirth)

5. **Smart Agent UI**: Hypertext Markup Language (HTML) webpage that displays patient information and calculated infusion rates and provides the ability to select and send calculated infusion rates to the infusion pump

6. **Smart Agent Insulin Infusion Algorithm Java Library**: library that implements the JHH version of the Yale Insulin Infusion Protocol

**Description of the software flow.** The following numbers correspond with those appearing in the circles shown in Figure 2:

1. When the clinician initiates the Smart Agent software by selecting the link in the EHR, the system sends the current patient information to the SAWS.

2. The SAWS calls Mulesoft to get the patient BG laboratory results and calculate infusion rates. (Mulesoft refers to the Anypoint Platform, a hybrid integration platform that enables organizations to easily build and rapidly scale an application network of applications, data, and devices through application program interfaces [APIs] and integrations.) These laboratory results, along with the current infusion rate from the pump, are used to calculate infusion rates using the Smart Agent Insulin...
Infusion Algorithm Java Library. The SAWS returns the Smart Agent UI as an HTML webpage to the EHR-embedded browser.
3. Below the current patient data, the UI displays the new calculated infusion rates. Next to each calculation, the UI also allows the clinician to review the calculations. The clinician then will select and accept a calculated infusion rate, which sends a JSON request to change and set the infusion rate on the pump.
4. The SAWS processes this information by transforming the received infusion rate into an unsolicited transmission of an observation HL7 message (ORU^R01) and notifies the clinician that the selected dose was sent to the pump.
5. The SAWS uses the HL7Sender component to send this HL7 message to MedNet/Mirth.
6. MedNet/Mirth receives the HL7 and serves as a proxy to forward commands to change the infusion rate in the pump.
7. The infusion pump notifies the clinician to accept the new infusion rate. Once accepted, the pump notifies MedNet/Mirth that the infusion rate has changed.
8. MedNet/Mirth replies to the HL7Sender component using a general ACK HL7 message (ACK^A01) with a status of Application Accept (AA).
9. In addition, MedNet/Mirth triggers an event change that sends an observation report HL7 message (ORU^R42) with infusion pump data to the HL7Receiver component. This component responds with a general ACK HL7 message (ACK^A01) with a status of AA to MedNet/Mirth.
10. HL7Receiver sends an HL7 ORU^R42 message to the Service Controller to transform this HL7 message into JSON that contains current pump data.
11. The SAWS records this JSON in MongoDB and sends a message back to the Smart Agent UI in the EHR-embedded browser confirming that the pump infusion rate has changed.
12. The SAWS enters the confirmed pump setting back into the patient record in the EHR.

A Unified Modeling Language sequence diagram (Figure 3) shows the steps in which Smart Agent obtains patient data from the EHR and calculates the new infusion rate. Several calls are made through MuleSoft to the EHR. Data are gathered, parsed, and sent to the JHH Insulin Infusion Algorithm. These results are then combined with data from the EHR and sent to the user via HTTP (HyperText Transfer Protocol) in an HTML page.

Proposed Workflow Description
The researchers propose that control of BG levels, which currently relies heavily on manually determined insulin infusion rates, should transition to an automated decision support algorithmic control. In this scenario, the clinician checks the patient periodically on a schedule defined by the protocol. The periodicity is identical to the lab checks of blood/serum glucose obtained using a point-of-care glucometer reading or a laboratory reading. The clinician then opens the Smart Agent link within the EHR medications summary page. Current insulin infusion dose settings and current and previous BG levels are transferred to the Smart Agent system from the EHR.

Smart Agent then will compare the current pump dose rate with the latest value entered in the EHR. If the dose rates do not match, Smart Agent will return an error and allow the clinician to update the EHR with the latest insulin dose rate. If the rates match, Smart Agent runs the data, including the available BG level(s), through the IV insulin infusion algorithm and presents the appropriate new insulin dose levels. To verify how the Smart Agent determined the new rate, the clinician may select the information button and review the insulin algorithm determination method.

The clinician then chooses one of the available insulin dose levels. The new insulin dose is sent to the insulin infusion pump, and Smart Agent indicates that the new dose was
sent to the pump. The pump will not automatically start the new dose upon its delivery from Smart Agent. The proposed dose is presented to the clinician on the infusion pump UI. The clinician has the opportunity to accept the new calculated dose or clear the new suggested dose and continue the current insulin infusion dose. If the clinician accepts the dose, Smart Agent will display a confirmation message to the clinician. The new insulin dose and the associated BG reading are stored both in the Smart Agent memory and in a worksheet of the EHR. Smart Agent can now be closed.

Results

Interoperability

Using the SAWS, the team was able to connect the JHH Epic EHR system to a commercial medication infusion pump. Information then was automatically translated in real time between these two subsystems. The SAWS used HL7 and JSON to communicate with the pump and the EHR.

Safety, Efficiency, and Accuracy

The SAWS removes the need for a manual determination of insulin infusion updates. The SAWS Java library has been validated to return the correct dose setting for every combination of current and past BG readings and current insulin dose rates. The web service also exchanges information with and monitors the infusion rate of the infusion pump. When the web service receives a request to calculate a new infusion rate, it sends the latest data to Smart Agent. If for any reason the infusion rate is changed on the pump but not updated in the patient EHR, the SAWS recognizes the mismatch and notifies the clinician upon any future call of the web service from the EHR.

Initial Limitations

In today’s interconnected world, the Internet of Things allows for great advancement in the movement and manipulation of digitized health data and remote control of medical devices based on these data. As such, robust security measures must be instituted to ensure that health data are not compromised and nefarious control of connected devices does not occur. Ensuring the security and protection of these system has been sought through the publication of numerous reports and the efforts of various organizations. However, JHU APL did not undertake the development of infrastructure or software to protect the Smart Agent system. With the completion of the demonstration of interoperability, an obvious next step is to develop and demonstrate this security.
The SAWS is not limited to one infusion pump–EHR pairing. Although the work that JHU APL executed while integrating the development instantiation of the EHR to the infusion pump was performed using only one patient record and one infusion pump, this was an experimental simplification. The SAWS is capable of receiving and processing multiple patient data inputs through the insulin infusion algorithm and then sending the new infusion rate to the infusion pump associated with a selected patient. This was not demonstrated in our work because only one infusion pump was available.

Analysis
Interoperability
The Smart Agent communicates with the insulin infusion pump translator, MedNet/Mirth, using HL7 messages in Integrating the Healthcare Enterprise–based profiles. (Note: Medication administration data transfer is not activated in the EHR at JHH.) The HL7 messages are then translated by Mirth into a proprietary XML (Extensible Markup Language) message to communicate with the medication infusion pump. On the other side of Smart Agent, data messages, packaged in JSON, are sent to and received from Mulesoft, the communication interface with the EHR. (Note: SAWS is working with the development instantiation of the EHR at JHH.) The Mulesoft service, which also communicates with the EHR using JSON, has been configured to send and receive simple messages to and from outside APIs using a RESTful format, which is familiar to most programmers. The Mulesoft interface acts as an intermediary and provides security so the information technology department at JHH can control what information is transferred to and from the EHR.

Safety, Efficiency, and Accuracy
Although JHU APL vigorously tested Smart Agent, the Armstrong Institute is performing clinical human factors and safety testing and will publish its results separately.

Recommended Future Research
The following endeavors may be considered by the Armstrong Institute and JHU APL or other research groups:

1. The technical aspects of the SAWS have been documented. To acknowledge the clinical usefulness of Smart Agent, the Armstrong Institute at JHMI will perform a human factors safety and effectiveness study of the SAWS from within the EHR.
2. As previously mentioned, nefarious actors seek to compromise all forms of web-based software. JHU APL did not consider the security of the SAWS because it will be located and operate behind the JHMI firewall. To ensure the integrity of Smart Agent, scientists should analyze and incorporate the necessary security features into the SAWS.
3. Our work has demonstrated the effectiveness of the SAWS to change the insulin infusion rate for one patient/pump combination. Further research can be performed in which the system is operated for multiple simultaneous patients (i.e., multiple bedside workstations accessing Smart Agent from a variety of patient locations).
4. To avoid any interference with the operational instantiation of the EHR, the team worked exclusively on the development version of the JHH EHR. After performing the safety and effectiveness study and validating Smart Agent, implementing the algorithm within the operational EHR should be considered.
5. Computer software scientists may need to explore programming the SAWS to send and receive data exclusively using HL7 version 2.0 or the Fast Healthcare Interoperability Resources (FHIR) standard, which became available in the JHH EHR in fall 2017, to demonstrate interoperability and drive development toward the healthcare industry standard. JHH uses Ensemble and Mulesoft to translate HL7 and FHIR messages to and from the EHR.
6. Because Smart Agent collects and stores BG values as a result of changes in the rate of insulin infusion, a much more encompassing analytical clinical study of the automated Yale Insulin Infusion Protocol should be carried out using the Smart Agent system.
Conclusion
A semiautonomous system that allows a clinician, operating from within the EHR, to interact with an algorithm on a web service, Smart Agent, was developed to automatically calculate the next infusion rate to be delivered to a patient. We assert that healthcare would be safer, more efficient, and yield better value if Smart Agent–type algorithms were more ubiquitous. Our vision is that healthcare technology will adopt a family of smart agents that can automatically perform tasks more safely than humans. This automation would unburden clinicians from the workload currently required in the absence of smart algorithms and interoperable systems.

Acknowledgments
To the following individuals for their vital contributions: Mark Romig, MD, assistant professor, Anesthesiology and Critical Care Medicine, JHH; Cynthia Dwyer, RN, senior research nurse, JHH; Michael Rosen, PhD, associate professor, Department of Anesthesiology and Critical Care Medicine, JHH; Aaron Dietz, PhD, health systems specialist, Department of Veterans Affairs; Noah Barasch, MS, quality and innovation project administrator, Armstrong Institute for Patient Safety and Quality; and Peter J. Pronovost, MD, PhD, chief clinical transformation officer, University Hospitals.

References