REDOX^

Direct Scheduling Strategy

Understand how to best utilize appointment information.

Executive Summary

Scheduling information is a crucial part of a patient's journey. While administrative in function, appointments will in most cases form the basis of a patient's outpatient clinical care documentation. Beyond that, the logistics surrounding appointments can also be clinically relevant. Knowing when, why, and with whom a patient is being seen is a powerful set of data for applications to help guide, ease, and improve a patient's interactions within a healthcare organization.

Additionally, scheduling information is important from an administrative and organizational management perspective. Efficient and effective utilization of providers' time has direct implications for organizational revenue, patient care, and physician happiness. Ensuring that physician time is scheduled and used appropriately and efficiently will ensure that each patient is receiving the best and most thorough care possible.

To this extent, sending, receiving, and querying scheduling information is a common request from digital health applications. In this whitepaper we outline the strategies and information that application developers can use when architecting their solutions pertaining to scheduling information, as well as some of the advantages and drawbacks of different approaches.

Relevant Data Models

There are two types of Redox Data Models critical to the success of a direct scheduling workflow. The first is the scheduling model that actually transmits the scheduling information, and the other is a patient identity model that helps the third party scheduling system maintain and map patient identity to the correct patient identifier(s) within the connecting healthcare organization.

Scheduling Data Models

Scheduling

The Scheduling Data Model is used to communicate the information surrounding a patient's appointments. At certain events in the normal scheduling process (scheduling, rescheduling, arrival/check-in, check-out), EHRs will send scheduling messages to keep the appointment status, date/time, scheduled providers, and scheduled resources in sync with the third party scheduling systems. Typically the most common way for EHRs to transmit these scheduling notifications is through a standard SIU HL7v2 feed, although Redox could support receiving this data using a number of exchange mechanisms, such as polling API endpoints. Redox can either push this data in real time as it becomes available or or persist it and provide a queryable endpoint to connecting applications who prefer to pull scheduling data as needed.

SurgicalScheduling

The SurgicalScheduling Data Model is used to receive surgical case information, such as scheduled time and location, staff assigned to the surgery, case and procedure details, etc. This may come from an OR-specific system or from the same scheduling system as non-surgical appointments. This Data Model is typically used for just the actual OR case that's been scheduled – if you want to receive pre and post-op office appointments, see the Scheduling Data Model. Similarly to the Scheduling Data Model, the most common way connecting healthcare organizations transmit this surgical case information is over an SIU HL7v2 feed, although again, Redox can support additional exchange mechanisms that may be more available or better supported by the source system.

Patient Identity

Patient Identity is a complex topic and is relevant to almost every Redox integration workflow. For more detailed information beyond the below, check out our Patient Identity white paper.

PatientAdmin

The PatientAdmin Data Model is the most common feed used across Redox customers and is something we would recommend you require if you persist patient data in your application. Over 90% of our installs use PatientAdmin in some form.

This is the primary method for you to stay in sync with the health care organization regarding the patient's documented demographics, insurance information, and the health system's patient identifier (usually called a medical record number or MRN). PatientAdmin is the only webhook-based feed that communicates changes to this type of patient information, usually through a push notification generated from the health system's ADT HL7v2 interface.

PatientSearch

If a connecting application would prefer a RESTful approach to verifying patient identity and obtaining patient data, the PatientSearch Data Model may also be an option. Applications using this model will need to have key patient inputs, such as demographic data and/or the patient's Medical Record Number (MRN), and the synchronous query response would return the patient's identifiers, including the MRN, along with their full demographics. Note that only some connecting EHRs will support a demographics-based query without an identifier as a key parameter, although Redox can leverage Data on Demand to support the PatientSearch model at any site with an active ADT interface.

Patient Enrollment Event

Summary

Enrollment is a powerful concept to know and understand when architecting your integrations with an EHR. Deciding how and when the right patients are added to your system at the right time can lead to better workflows and higher usability.

Scheduling as used as a patient enrollment event is the simplest way an application can be integrated with an EHR. When an application is concerned primarily with the outpatient setting, scheduling information can represent a unique event to enroll the patient into the application's database, allowing the application to add it to its worklist or relevant patient subset.

For instance, if you were developing a dermatology application, you could use the Scheduling.New event that is triggered by the creation of a new appointment in a dermatology department to add that patient to your database, or you could query Scheduling.Booked to obtain the list of all of the patients scheduled during a given time frame. Similarly, if you were creating an iPad application for patients to fill out questionnaires prior to orthopedic consults, your application could receive the Scheduling.Modification message triggered on the arrival of a patient for an orthopedic consult visit.

Supported Data Flows

Redox can support your application's patient enrollment needs through the Scheduling.Booked query or through the following event-driven Scheduling transactions:

- Scheduling.New
- Scheduling.Modification
- Scheduling.Reschedule
- Scheduling.NoShow
- Scheduling.Cancel

Additionally, through our engine filtering capabilities, Redox can help ensure that you only receive the patient subset relevant to your application's workflow needs, minimizing data storage needs and PHI exposure.

After initial enrollment, many applications look to augment their picture of the patient with Data Models such as PatientSearch or ClinicalSummary.PatientQuery. Any Redox data model enabled for Data on Demand can be queried for information after your initial

enrollment regardless of the capabilities of the EHR, though some Data Models may only be available via a webhook-based push.

Appointment Consumption

Summary

Beyond use as an enrollment event for your relevant patient population, scheduling information is useful for both administrative and clinical purposes. Applications can show appointment information in administrative capacities, such as for patients to understand their planned activities at a hospital, for providers to help them visualize their workload, or for other users to optimize organizational efficiencies. Users can also derive clinical meaning from appointment information, using data elements like the reason for visit, related diagnoses, or appointment notes to understand a patient's problems and care.

For instance, a patient portal application may enroll all patients when they are created via the PatientAdmin Data Model but may supplement this information by querying for scheduled visits through Scheduling.Booked or by receiving an event-driven Scheduling feed in order to show the patient their upcoming appointments.

Supported Data Flows

Redox can also support your application's needs through the following query-driven Scheduling transaction:

- Scheduling.Booked

This query based event uses criteria, such as the patient's MRN and a date range, to find booked appointments and return relevant information corresponding to that criteria.

Redox can also support your application's appointment consumption needs through the following event-driven Scheduling transactions:

- Scheduling.New
- Scheduling.Modification
- Scheduling.Reschedule
- Scheduling.NoShow
- Scheduling.Cancel

As with enrollment, through our engine filtering capabilities, Redox can help ensure that you only receive the patient subset relevant to your application's workflow needs, minimizing data storage needs and PHI exposure.

Direct Scheduling

Summary

Direct scheduling systems allow patients or users of an external application to schedule appointments without interacting with the health system's scheduling staff.

The goal for a direct scheduling application is to identify when providers are available to be scheduled and then fill those slots. The challenge is ensuring the application has a complete picture of the provider's schedule prior to filling those slots. Just because a provider doesn't have an appointment at a given time does not mean that the time slot can be filled by an appointment from our customer - the provider may only be seeing certain types of patients, may be out sick, or might be on a break golfing. For our customers, this poses a significant challenge - if they receive an appointment cancellation, does that mean the patient canceled or that the provider is no longer available?

In these scenarios, the health system's scheduling system will typically be considered the ultimate "source of truth". Any other system allowed to schedule appointments will need to close the loop with that application to avoid double-booking and other conflicts.

Supported Data Flows

- Direct Scheduling with Available Slots
    - In this architecture, the application queries the EHR using the Scheduling.AvailableSlots and associated response events to find available date/time slots that a given provider or resource can be scheduled.
    - The application can then use event-driven Scheduling messages (New, Modification, Reschedule, NoShow and Cancel) to schedule a patient with a provider for the ascertained date/time.
    - As this is not supported through traditional HL7 interfaces, in order to use this architecture, it is required that the EHR vendor offers the Slot FHIR resource or a vendor-specific API that exposes Available Slots.
    - The Redox Data on Demand feature will introduce support sometime in 2020 for available slots queries where they are not natively supported by the connecting EHR, assuming the availability of certain pushed data feeds. Your Redox representative can keep you updated on timelines related to this feature.
- Direct Scheduling with Agreed Upon Blocks
    - In this paradigm, a provider defines a block of time on their schedule that is held for external appointments (i.e. 2-4pm on Tuesdays and Thursdays).
    - If the time slots are still open either the day of or the week of, they auto-release for same day appointments or for anyone to schedule into.
    - The application can then use event-driven Scheduling messages (New, Modification, Reschedule, NoShow and Cancel) to schedule a patient with a provider for the ascertained date/time.
    - This is the most common workaround for external scheduling in the event that an Available Slots API is not offered by an EHR, in that it requires only working agreements and minimizes the need for additional interfaces.
- Direct Scheduling with Daily Availability Extract
    - The EHR provides a daily extract of the provider templates (the definitions of when and for what the provider can or can't be scheduled).

- This is provided to the application as a flat file or via the Scheduling.PushSlots API.
- By also consuming a Scheduling feed, the application has a picture of what slots are available, although any "day of" changes, like provider sickness, will not be included.
- The application can then use event-driven Scheduling messages (New, Modification, Reschedule, NoShow and Cancel) to schedule a patient with a provider for the ascertained date/time.

In addition, applications can supplement any of the above strategies with the use of an asynchronous confirmation message.

Direct Scheduling with Asynchronous Confirmation

- When the application schedules back into the EHR using event-driven Scheduling messages (New, Modification, Reschedule, NoShow and Cancel), successfully booked appointments are rebounded back to the application. Appointments that fail to be booked due to errors do not have a response to the application.
  - If the application receives back an echoed version across the Scheduling feed from the EHR, the appointment can be considered successfully booked.
  - If not, the appointment should be rebooked.
- This paradigm is recommended in combination with the “Direct Scheduling with Agreed Upon Blocks” and “Direct Scheduling with Daily Availability Extract”.

Conclusion

While challenging, effective use of patient and provider information regarding appointments can be valuable in creating innovative healthcare applications. To further these efforts, Redox offers a variety of ways for your application to obtain, utilize, and create scheduled appointment information. The Redox platform can accelerate the development and

distribution of healthcare technology with a full-service integration platform to securely and efficiently exchange different data sets, such as scheduling data. Healthcare organizations and technology vendors connect once and authorize the data they send and receive across the most extensive interoperable network in healthcare. Redox exists to make healthcare data useful and every patient experience a little bit better and helping to facilitate direct scheduling is just one of the ways we do that.

***Review Developer Documentation** about scheduling and start building applications that seamlessly integrate appointment data with any EHR.*

***Contact our Solutions Team** for any additional information.*