



Copyright © 2020 by the Construction Financial Management Association (CFMA). All rights reserved. This article first appeared in *CFMA Building Profits* (a member-only benefit) and is reprinted with permission.

BY CHRISTIAN BURGER

Integration Strategy: A Primer for Today's Construction Company

Over the past 20 years, the need for solution integration strategies in construction has shifted from convenient to ideal to *critical*. However, the construction industry has not kept pace with technology to facilitate these integrations. Many companies continue to manually reenter data in multiple systems, often after manually manipulating the data in a spreadsheet or by performing a nightly batch upload using a CSV file that is produced in one system and consumed in another.

This article explores the changes in software applications used within the construction industry, the different types of integration approaches available, and the associated benefits and drawbacks of each. It will also discuss considerations when developing a strategy for your company as well as steps to help begin deployment. See the next page for a look at some key concepts that will be addressed throughout the article.

The Need for Better Integrations Enterprise Resource Planning

While construction companies have utilized fully integrated enterprise resource planning (ERP) systems with many software applications or functions available, those rarely included estimating, time capture, and scheduling. Only recently have they included a competitive version of a project management system and have all but given up on a robust human resource (HR) application; service management is now often a third-party application as well. Think of ERP as a hub and every application that has a financial transaction associated with it as the spokes; the hub is getting crowded and those integrations need to be controlled.

Meanwhile, the industry is awash in new applications that handle tangential functions to ERP (e.g., scale ticketing, timecard processing, and HR), many of which did not anticipate integrating with ERP or any other solutions, leaving contracting companies with as many as 5-10 applications that act as data silos in the organization.

Complications in the Cloud

There has been a substantial shift to cloud-based applications, which outsource the infrastructure away from on-premise

data centers and applications. Rather than having control of the applications and databases from within your IT department, control has shifted to the service providers, hosting companies, and software providers that are very selective about how they allow access to their applications and data (because all of their customers' data is often in one database (i.e., multi-tenant)). This means that integration approaches that once worked well with programs and data in your building are no longer viable (or at a minimum are more challenging), though some of these vendors are providing openings to enter data or extract data out through APIs.

Complicating matters is the frequency with which software vendors provide new features or updates to their applications. Historically, this was an annual or semi-annual event and was not as disruptive as it can be today with frequent updates being pushed out by vendors, sometimes even monthly. These updates have the ability (or tendency) to break integrations that were written to connect one application to another.

So, after each new update, the integrations are checked in a "test" environment (a copy of the production environment of a given system that is used expressly for testing changes in processes, upgrades, and custom programming) before promoting the patch or upgrade to your production environment. This allows for thorough testing without interfering with the live production environment.

Using a cloud-based application rarely provides this luxury. Often, vendors shut the application down at night, install the upgrade, and the next morning you have new features. This makes maintaining custom applications more difficult.

Time to Prioritize Integration Technology

There are a number of reasons why technology and an integration strategy should be a moderately high priority in your organization:

- IT departments remain understaffed and don't have sufficient time for maintaining or developing integrations on a point-to-point basis.



- Your personnel's time is valuable and costly. They should not have to monitor batch upload routines, fix errors on a daily basis, or manipulate data in spreadsheets to ready the data for import.
- Your operations and finance personnel are far more data-driven and need accurate information on a timely basis. They cannot wait for batch uploads that are manually performed and could be late or inaccurate.

These reasons, coupled with the increased use of point-solutions and cloud applications as well as the availability of better technology to develop scalable integrations, make it a compelling time to address this lagging part of your systems architecture.

Why Middleware

While not the only resource available for integrations, middleware is perhaps the most underutilized utility in the industry and is becoming standard for mid-size and larger contractors. Some solutions have adapters built for the common functions in project management and ERP, while others are more powerful yet generic (e.g., they don't distinguish a time card from an RFI when you install the solution).

Some primary features to consider when deciding on middleware include: scheduling, error handling, audit trail, data validation, storing data vs. transient data, sequencing, easy-to-use interface, workflow/approval, logic (for amending data), data transformation, multi-layered security, encryption, and document caching (i.e., for better overall performance, during transformation, and memory processing). Your IT department should have a heavy influence on this decision.

While middleware is not as easy as "drag and drop" programming, it also isn't as difficult as coding. A competent IT person or business analyst could be trained on how to use the platform or tool set to develop integrations. This is not a system where you should expect the provider to stay involved permanently, though they will gladly provide support to their customers.

Some degree of middleware should be a part of most mid- and large-size contractors' arsenals when multiple systems with multiple high-volume integrations exist. But just buying middleware may not be sufficient either, as a vendor's platform state can vary from conducive to challenging.

Key Concepts

Validation

Validating data ensures that the data moving from the source system to a target system is valid or expected. For example, you cannot move invoice data from one system to another if the target system does not recognize the vendor or job ID. Consequently, all the tables in a target system must be up to date with new customers, vendors, jobs, etc., before new data is posted. Otherwise the system will reject the record (or the entire batch) until the error is fixed.

APIs

Simply put, an application programming interface (API) is a small piece of software code, provided by one software vendor, that allows other software developers to connect to and transfer data between another application and the intended target system in an automated and efficient manner. An API is a more secure approach to moving data from outside an application into its tables since writing directly to a table is *never* advised.

Middleware

Middleware is a software utility that acts as a bridge between applications, data, and end users. This is not a commonly

understood technology in construction and is underutilized as an integration strategy. This is slowly changing as source and target vendors are making their systems more open to integration, and the cost of these utility solutions is coming down.

While middleware might be excessive for companies with only one or two solutions to integrate and a modest number of transactions, the value increases when the number of integrations, transaction volume, and frequency is high *and* the sequencing has to be carefully orchestrated.

RPA

Robotic process automation (RPA) is also a small amount of code specifically designed to sign into an application (target) and enter data from a file that was provided from another system, as though it were a user.

RPA is a good approach for high-volume predictable data that rarely has validation errors and should be used where the target system does not offer much in the way of APIs or other importing solutions. (For more information, read "Transform Your Construction Operations Through RPA" in the March/April 2020 issue.¹)

Moving Your Data

In general, there are a few primary approaches to moving data safely from one solution to another. Exhibit 1 focuses on each option's efficacy and rates it on a scale of 1-10 (10 being ideal).

Most companies do not settle on a single approach before moving forward since much of the decision depends on the various vendors, what they allow, the integration technology provided, cost, risk, and urgency.

Integration Process

Exhibit 2 on the last page shows the process for an integration to work smoothly using an example of a standalone HR solution creating an employee Masterfile in a separate payroll application. This would be an example of a well-integrated point-to-point integration or a middleware enabled approach. It demonstrates each step and its importance in the process, although this is not unique to payroll and HR integrations.

Strategy & Deployment

Even though your integration strategy can be deployed slowly and in stages, it is best to have an overall scalable

and supportable approach decided before you begin. This prevents you from developing one approach and then backing up and redoing some work because you discover a better approach later.

The first step is to develop an integration requirement profile in your organization. This could be done simply by completing an integration schedule like the one seen in Exhibit 3 on the last page to help determine how many integrations are in your organization and the relative importance and priority of each.

This will also give you a chance to learn more about the systems you have in place, their integration capabilities and shortcomings, and what issues are associated with the current integration method. This provides a more holistic view of your situation and helps you gauge the value of the solution.

Once you have this completed, you can begin to get an overall picture of the requirements, risks, and values of developing an integration strategy. You don't have to have a single strategy, but something coordinated and planned is better and easier to maintain.

Exhibit 1: Efficacy Chart

Approach	Description	Efficacy
Fully integrated suite (base-line comparison)	Using a fully integrated suite means no data has to be moved in or out.	10
Vendor-provided integration	Similar to a point-to-point integration, this solution is specifically developed for two end-points, typically by one vendor that wants to make it easier for a customer to move data from its solution to another predictable endpoint (e.g., vendors develop and maintain an integration for timecards over to payroll applications in various ERP solutions).	8-9
Middleware	This third-party solution sits "in the middle" of a series of applications and acts as a data broker, moves data from one application to another, handles error trapping, and provides messaging and an audit trail. If necessary, it also has the ability to transform data before import.	8
Point-to-point	A custom-built integration that uses various utilities from one application to another. This can be maintained internally or by a vendor for a fee. This is generally efficient provided an error-handling process exists and doesn't break on update. This is also dependent on both vendors having appropriate technology available (e.g., APIs).	7
API and web-service	When available, these technologies provide a controlled and accurate way of importing data into an application's tables or extracting and sending data to another location.	6-7
Batch upload	This is the manual approach of getting a CSV file from one application or database and uploading it into another application. Many estimations to job cost integrations are handled this way. Error trapping is manual, the effort to upload is manual, and the timing is variable.	5
Rekeying	This is the worst-case scenario. Although rekeying data from one system to another likely happens frequently, this approach is obviously problematic.	2



Initially tackling some low-risk integrations will help you practice before turning to large-scale complex integrations. Generally, some level of cooperation from both vendors will be needed unless one or both have a full library of published APIs with documentation that you or your developers can use.

You also must assess whether you can place a monitoring agent on the server where the target application lives in order to push data in. Some multi-tenant cloud and even hosted applications resist this.

You also need to be sure you have a test environment available for the target solution to run multiple tests (including error trapping) from your source to the target solution, rather than running tests in your live or production environment.

And, depending on the vendor's update schedule and your ability to control the loading of updates, make sure to retest integrations in the test environment after an upgrade is loaded to ensure the update did not break the integration.

Conclusion

With so many new independent applications operating separately in a typical environment, companies must develop a more secure, efficient, and scalable approach to moving data from one system to another. If you're unsure as to whether an integration strategy is necessary, ask yourself these questions:

- 1) Are people in the organization babysitting manual extractions and uploads from one system to another?
- 2) Are people in the organization spending time massaging data in spreadsheets in order to prepare it for import to another system?
- 3) Are people questioning the accuracy of reports and inquiries due to lack of data that is otherwise available in another system?
- 4) Are people in the organization spending time reentering data manually from one system to another on a regular basis?

If you answered yes to any of these, take a step back, assess the integrations you either have or need, and then begin to evaluate the integration capabilities of your core systems. With a completed integration schedule in hand (Exhibit 3), enlist the help of your IT department to vet the various solutions available and develop the best strategy – solving some of the inefficiencies in the organization could easily have a quick payback. ■

Endnote

1. www.cfmaonline.net/cfmap/20200304?pg=28#pg28.

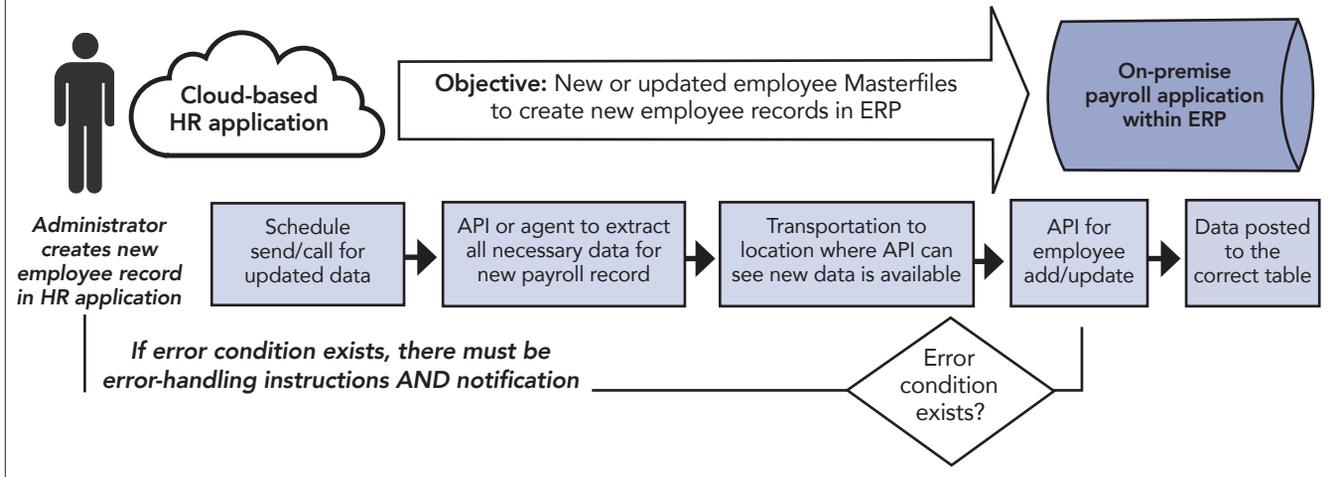
CHRISTIAN BURGER is President of Burger Consulting Group, an IT consulting firm in Chicago, IL. Christian has been a member of CFMA for 25 years, and he has been involved at both the local and national levels. He has written for *CFMA Building Profits* and presented at the national, regional, and chapter levels on technology.

Phone: 312-651-4172

E-Mail: crburger@burgerconsulting.com

Website: www.burgerconsulting.com

EXHIBIT 2: HR to Payroll Integration Example



Step	Description/Action
New Record Created	This happens normally within a source application by a user entering new data.
Record Data Extracted	Either the source application must realize that a new record was added and create a copy of the record, or a scheduled process should be run to extract all new records.
Record Data Moved	The extracted data must be moved to a place where the target solution or an agent for the target solution can access it. No validation has yet occurred.
Notice to Payroll Agent	The target solution identifies that a file containing one or more records is ready for import.
Payroll Agent Reviews Record	API or other routine ingests record for validation. It checks all necessary fields to ensure they are in the right format and that the Masterfile data is accurate (e.g., employee classification). This is where kick-outs occur and either a single record is suspended or the entire batch is held up.
API Imports and Posts Record	Once the validation clears, the batch is released into the target system as though it had been entered manually. A message may be sent to an administrator indicating that one or more records have been updated.

EXHIBIT 3: Integration Schedule Example

Source Solution	Extract Available?	Target Solution	Integration Tech Avail.	Data to move	Volume of data	Frequency of update	Priority/Criticality
Estimating	Yes/No	ERP Job Cost	APIs	Budget records	Moderate	Moderate	Medium-high