White Paper:

Today’s Challenges in Automated Scheduling… and What the Future Holds

FIELDONE
ENTERPRISE FIELD SERVICE

Fall 2012

FIELDONE
ENTERPRISE FIELD SERVICE

If you spend your day wrestling with the complex logistical challenges of field service management, then you are aware that most companies — small, medium and especially enterprises — are implementing technologies to relieve the stress, errors, and wasted expenses that had been an accepted aspect of manual scheduling. The core solution is, of course, **automated routing**.

Take the following illustration:

Following a brief citywide power outage, Dispatcher Dave gets a call from Mercy Hospital, reporting an electrical malfunction in the Pediatrics ward. Of 160 techs in the field, about 71 have the appropriate skills and most probably have the right equipment with them. So he's got a long list to choose from.

But now, who to send? One of his quickest electrical specialists is **Robby**, but Robby's currently on his way another job. Next is **Lenny**, who is available, but has another job scheduled for an hour from now. Dave has to check to see if that job can be delayed if needed. Then there's **Mel**, who is about to leave his home for a non-urgent maintenance visit which *can* easily be postponed — but he's supposed to meet **Mark** at the first job site. So because Mark wouldn't be required at Mercy, he would need to be re-assigned. If Dispatcher Dave *does* re-assign him, the most logical job would be to a site where he'd be replacing a part in an emergency generator that would need to be online by the time he leaves, regardless of time — and his overtime rate is higher than average. And to make it more interesting, Dave has no idea if the required parts are standard components. **Robby**, who just left the office, could theoretically swing back and take parts from the warehouse; **Lenny**, in the field, may need to return to the office to pick up the parts, and Mark, leaving from home, will definitely need to do so. But how *far* is Mark's home, compared to Lenny's current location and time to completion? And so far, Dave has only considered four techs…

And let's keep things interesting: while Dave is considering his options, a call comes in for a similar electrical problem in a mall, clear across town. This customer is under SLA, so without having quite solved the Mercy Hospital challenge, Dave begins strategizing about his parallel call, but at a higher level of urgency. *And he's only been in the office for six minutes…*

You don't need a degree in Psychology to know that the human mind cannot absorb, manipulate and make sense of thousands of pieces of *static* data, not to mention *dynamic* information that flows hourly through a field management system. Variables like a tech's skills and current location, parts on hand, team makeup, client location, lunch hours, vacation time, customer availability, start/end locations, client contracts and dozens more all mean that a human-driven process is, by definition, sub-optimal in three ways:

1) Under time pressure, people look for a solution that "works" rather than one that's ideal or the most cost-effective. In short, dollars leak on every job assignment.

2) Balancing customer and company priorities leads to one side "sacrificing" for the other — expensive workflow for the company and long wait times at the client site.

3) The impact of the Domino Effect. Even with a carefully built, well-organized schedule, a single unexpected event can require a complete reworking of the day's assignments…triggering and amplifying the considerations. And it happens over and over.

> **"You don't need a degree in psychology to know that the human mind cannot absorb, manipulate and make sense of thousands of pieces of static data, not to mention dynamic information that flows hourly through a field management system."**

The science of automatic routing — the central component of any comprehensive field service management software solution — is based on algorithms. A computer considering tens of thousands of potential combinations can identify the optimum solution — and then do it all over again (without complaining!) a moment later if circumstances change. The moment a new variable appears — as basic as a tech out sick, or as subtle as a 20-minute delay in traffic — the combinations are shuffled and the process starts again, re-evaluating every single combination from scratch.

It's magic, humming along quietly "behind the curtain," in the background.

If only it were that simple.

The reality is that your world of fluid, often unpredictable service tech routing, even complex algorithms often cannot do the job perfectly in the time required. Here's why. In what's called a Linear Problem, the decision about one event does not affect any future events. With all the variables static and available for the analysis, it's easy to rule out whole batches of options without examining them too closely (e.g., Tech A isn't qualified and Tech B is working at another location, so they are not considered. Remove 700 scenarios in which they are involved). In these cases, we never need to literally examine each and every option.

> **"Choosing any particular solution sets up new variables and creates a completely new set of options for the next decision."**

For a *Non*-Linear Problem, though, facts are always in flux. The simplest example is the tech whose jobs for the remainder of the day may be affected by the location and time spend on the first day's job. A more complex example may involve traffic changes throughout the day that change the time it takes to reach the next job; taking historical data into account can change the optimal plan, possible sending a tech further away, but earlier in the day to avoid the traffic. This type of problem can only be solved using what's called Brute Force — assessing every possible combination. As you can imagine, with dozens and dozens of factors, this process can mean reviewing and prioritizing hundreds of thousands of scenarios. This is not a particularly efficient.

Software developers work with heuristics (with names like Local Search, Hill Climbing, Tabu Search, Tree Search, Simulated Annealing, and more),where the goal of the process is to complete these comparisons in minimal time, requiring minimal computing "horsepower". While it is, in broad strokes, a Brute Force approach, we don't attempt to necessarily reach the *single, very best solution*, but rather to get ***one*** of the very best (from a group of "finalists" with negligible downside dividing them) so that the team can act on it quickly and move on to the next challenge.

"Step aside, Human"? Not so fast...

There are certain areas of field service management that we can confidently leave to automated, computerized processes. For instance, client updates, alerts and confirmations can be sent handily using automated voice, SMS and email: "*Your tech, **Rob**, is en route to **492 Livonia Blvd** and should arrive at **10:32 am** to repair your **Air Intake Valve.***"

Likewise, automatic process or event triggers have evolved nicely to relieve humans of laborious tasks: A tech in the field registering a job as "Complete" on his mobile device triggers an invoice being compiled (complete with parts used) and sent out, the parts being re-ordered, the client record updated, and your ERP system kept in sync.

"With the number of fluid, dynamic variables in play for a large enterprise, it's virtually impossible to be right all the time."

But what's different about routing?

If you could simply plug the facts into the software and let it lead the way, you'd have an ideal situation: few mistakes are made, few people are required to work on the dispatching process, and you could sit back and watch costs drop and revenues skyrocket the day after implementation of such a system.

But we humans have a long history of "suspicion" of computers taking control. (Think "computer-takeover" classic movies like *2001, War Games, Tron, Star Trek: The Movie, Terminator, Eagle Eye*). And for good reason — we've already accepted that, with the number of fluid, dynamic variables in play for a large enterprise, it's virtually impossible to be right all the time. Now we also have to recognize a further impediment to decision-making perfection: *we can't always supply all the facts.*

To draw an analogy, this is why on boarding a plane, you'll always see that there's a pilot flying it. Because really, he's not: These days, 98% of a flight is under the complete control of the plane's automatic pilot. And yet, we will always see pilots sitting in the cockpit to monitor every stage, to step in if required, and simply provide the peace of mind we all need, knowing an experienced human is ultimately in control. (Ironically, most flight tragedies are due to *human* error — but that won't change the policy.)

It's the same with field service management. Sure, we know the locations of both the tech and the client. We can often, using live traffic software like Google Maps or Waze, determine what kind of traffic stands between the two. We know the terms of an SLA, the pay grade of a specific tech, his skills, and the equipment he has in his truck. And if we know it, we can make sure our system's decision-making algorithm takes it into account.

But what happens when circumstances spin a different direction? Let's revisit those examples and find the challenges in the "computer-solves-all" formula:

- Tech location: What if his phone or tablet runs out of power or he's forgotten to turn it on, and thus isn't transmitting an accurate location?
- What if later today there's a parade scheduled to close a street that's currently (according to the computer), open and moving quickly?
- What if, despite the terms of an SLA, your company is worried about losing a particular client and wants to try to impress him? Or what if one client call is a bank with a vault security failure, while the other is a repair in one of a long panel of TV screens at a gym?
- What if, despite two techs with what's technically the same skills and qualification, we know that a specific one has worked on a job site before and although not specifically required, could get the job done more efficiently?
- What if a part is technically required for a job, but in the interest of completing the job in a timely manner, could be substituted or implemented later?

These are all facts that only human interaction and intervention can realistically be expected to incorporate into the decision-making process. Though every software provider works hard to include all common variables, there will always be many that are specifically *not* standard.

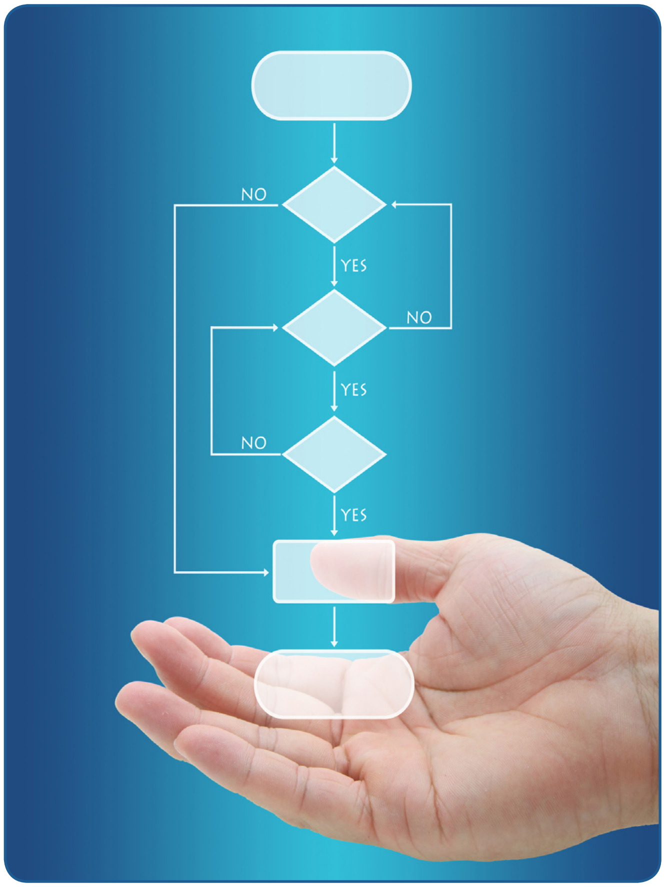
As such, what's the ideal solution?

You need a balance of automation **and** supervision of an experienced dispatcher, ready to step in and make judgment calls. All the algorithmic heavy lifting is surely the first step, and substantial technological development must continue to be invested in creating a system that yields the optimal workflow, based on the information available, in what feels to your team like real-time.

So after the algorithmic component, the next most important feature of your field management solution is the ease with which dispatchers (a much smaller team than you'd otherwise need, but still critical) can introduce new facts, modify work plans, and change assumptions — *all without 'frustrating' the system.* In other words, the software must recognize and "respect" the human interaction, and see these changes not as a challenge to its strategy, but a welcome part of it.

What's Next?

In addition to recognizing this human-computer partnership in optimizing daily workflows, field service technologies have some exciting developments ahead. Some of these are unique to the industry, and some adopt paradigms from others.



Beyond Maps — Providing a tech with a client location on a map is a basic requirement. Even better is an integrated point-to-point GPS system to help guide the tech to the job. The real evolution, though, is live, updated traffic maps (like that being offering by Waze or Google Maps, for instance) that can not only guide the tech as he drives, but also provide the software (and dispatchers!) back at the office, with accurate travel times, on the fly. Remember, a traffic snafu can delay a single job, or trigger the Domino Effect, delaying the tech's (and his colleagues') remaining jobs for the day. The tech needs to know it, but so does the system.

Dynamic real-world data can also be collected from train, bus and other public transportation options, providing alternative routes when practical. In large cities, these options, when feasible, are often faster and less expensive.

"Sandbox" Planning and Analysis — As we've agreed that most companies will want to maintain a level of human oversight in the automatic routing matrix, these individuals are faced with an interesting dilemma: they can't necessarily second-guess the computer, because they don't have the "mental horsepower" to contemplate the long list of alternatives to a particular fluid scenario.

A new feature on the horizon: a simulation system in which a dispatcher can experiment on *a snapshot of the current situation,* testing a "gut feeling" or urgent "no choice" scenario, without affecting the live system. Such a system is also useful for planning scenarios like taking vehicles out of the pool for maintenance, retiring (or firing) staff, or taking on more clients without adding to the current staffing resources. Using real-life data makes the simulation as close to reality as is possible.

This approach also provides for after-the-fact analysis: *"If I had sent **Jim**, who was further away with higher fuel costs, rather than **Joe**, who was nearby at another job and lined up for more, or **John**, who's on a higher pay grade and could arrive quicker than the other two, how would the rest of the scenario have played out?"*

Considering Costs — Most current routings systems are tuned to recommend a routing plan for maximum efficiency. Reduced "down time" for a tech between jobs, minimal driving to save both time and fuel, and most importantly, boosting customer satisfaction by assigning the right tech to the right job, at the right time. An additional, "meta-factor" currently being added to the mix is long-term *revenue potential.* This is a variable that's not about the tech himself, but rather about the client.

Imagine a situation in which you have two techs available for two pending jobs. Both are technically qualified, but one is clearly the more experienced and skilled… and paid accordingly. Now imagine that one of the two clients is paying your company on a per-job basis, whereas the other is on a more lucrative ongoing SLA. The extra cost of assigning the more skilled tech — even if he were further away or needed to return to the warehouse first for parts — would be justified by the long-term customer satisfaction consideration of the higher revenue-generating client. Needless to say, this balance between skill level, pay scale and the fuel/time costs is always a complex one to compute, and ongoing development will continue to make these automated decisions more and more precise.

Real Time Automatic Monitoring — Today, most urgent calls to a field service provider come from a stressed individual on site: "We've got a broken machine!" or "That system isn't working and business is suffering!" Though this sounds like a simple process, there are a number of weaknesses. First, failures are only noted when observed — which means that small problems can build up before they are noticed, but damage is being done. Second, failures occurring off-hours are left to be addressed during work hours. Finally, the reports are usually superficial, describing the symptoms, not the cause.

A number of companies offer hardware-based sensors that connect to system components, monitoring statuses 24/7. The minute a failure is detected, these systems typically send an alert to an internal system: an email, SMS or update to an intranet. It requires only a minor integration to have these sensors trigger a message to the field service company, triggering a job. In a sense, one computer is making a request of a second one, which then checks the schedule and staffing available and instantly assigns a tech to the job. Updates sent to the relevant personnel keep them up to date: *"Your GH-450 sent a failure message at 4:33 this morning. A Tech will arrive at 8:20 am to service it."*

Team assignments — With all the complexity involved with assigning individual techs, the challenge of sending teams raises an entirely more complex challenge. It's easiest to explain through an example: a boiler system in Location A needs maintenance, requiring a team of four to six techs. A similar job in Location B is located in a basement with space for only four. Location C needs only one tech for a smaller repair. With four techs available right now, do you assign all of them to A and then the B and C? It allows them to travel together, though it means Location B will have to wait to begin work, and C doesn't get attention, despite its need for only one person. It also means that you can't have one team finish early at A or B and move on to C. And how do you assign them to vehicles? What if the same equipment is needed for A and C, but only some of the team has the skills to use certain components? And finally, if job D appears unexpectedly, how many do you send, how quickly, and in what vehicle?

This added complexity to the more basic, classic scenarios is the next challenge for software developers.

Summary

Field service software has made dramatic improvements over the past decade, harnessing the power of new technologies borrowed from other realms, or custom-developed, in-house. This trend is only accelerating, sealing up the leaks in efficiency and revenue that had been (until recently) considered a part of the process.

Nonetheless, the ideal business workflow includes *both* the automated processes to tackle the analysis that the human mind cannot possibly handle, as well as methods for a dispatcher to step in and incorporate the wisdom and real-world variables that the computer cannot appreciate.

FIELDONE
ENTERPRISE FIELD SERVICE

About FieldOne

Our goal at FieldOne is simple: To help your service company increase efficiency and productivity by leveraging technology that was built for your business's specific needs. We take great pride in knowing that we assist enterprises around the world, across a broad range of industries. Our clients share a similar desire to better manage the complex nature of their service organizations, and nearly everyone across the company can benefit from our software - from the owners to the service managers, the warehouse crew to the administrative personnel, and the schedulers and dispatch managers to the technicians in the field.

Founded in 2001, FieldOne is an innovative, secure and scalable service management system providing field service companies with powerful tools to streamline their business processes. From up-to-the-minute technician scheduling and status information to on-site wireless data entry, complete inventory control and automatic invoice creation, FieldOne is an easy to use, all-in-one software solution.

Our innovative, secure and scalable software enables our customers to spend less time entering and managing data, freeing up valuable time for revenue-generating work. With FieldOne, organizations can more efficiently handle all of their day-to-day activities without wasting hours on duplicate work. Most importantly, they can spend more time helping their own customers and growing their business.

We have had the privilege to help in industries including property management, HVAC, medical and diagnostics, IT and technology, mechanical, janitorial, landscaping, contracting, plumbing, electrical, roofing, irrigation and security companies.

For more information or to learn more about the benefits of FieldOne, or how our solutions can help you grow your business, please schedule a free demo or contact us.

1 International Blvd., 6th Floor Mahwah, NJ 07495 1.866.918.8324
www.FieldOne.com info@fieldone.com