
The industrial internet

The barriers between software and the physical world are falling. It's becoming easier to connect big machines to networks, to harvest data from them, and to control them remotely. The same changes in software and networks that brought about decades of Silicon Valley innovation are now reordering the machines around us.

Since the 1970s, the principles of abstraction and modularity have made it possible for practically anyone to learn how to develop software. That radical accessibility, along with pervasive networks and cheap computing power, has made it easy to create software solutions to information problems. Innovators have responded, and have reshaped practically any task that involves gathering information, analyzing it, and communicating the result.

Something similar is coming to the interfaces between software and the big machines that power the world around us. With a network connection and an open interface that masks its underlying complexity, a machine becomes a Web service, ready to be coupled to software intelligence that can ingest broad context and optimize entire systems of machines.

The industrial internet is this union of software and big machines — what you might think of as the enterprise Internet of Things, operating under the demanding requirements of systems that have lives and expensive equipment at stake. It promises to bring the key characteristics of the Web — modularity, abstraction, software above the level of a single device — to demanding physical settings, letting innovators break down big problems, solve them in small pieces, and then stitch together their solutions.

The foundational technologies of the industrial internet are available now to anyone from big industrial firms to garage inventors. These technologies include: pervasive networks; open-source microcontrollers; software that can analyze massive amounts of data, understand human preferences, and optimize across many variables; and the computing power needed to run this intelligence, available anywhere at little cost.

Anyone who can recast physical-world problems into software terms now has access to the broad world of “stuff that matters”: conserving energy and reducing our impact on the environment; making our world safer, faster, and more comfortable; improving the productivity and well-being of workers; and generating economic opportunity.

Characteristics

The industrial internet¹ is an approach to bringing software and machines together, not a particular group of technologies. These are the principles driving its development.

Internet architecture and practice applied to industrial settings

The industrial internet isn’t necessarily about connecting big machines to the public Internet; rather, it refers to machines becoming nodes on pervasive networks that use open protocols. Internet-like behavior follows: machines publish data to authorized recipients and receive operational commands from authorized senders.

Think of the difference between an airplane built 40 years ago and a modern design like the Boeing 787. Older airplanes have direct linkages between systems — from the landing-gear switch to the landing gear, for instance. Newer airplanes use standard networks, in which the landing gear is a node that’s accessible to any other authorized part of the system — not only the landing-gear switch, but also safety, autopilot, and data-logging systems. Software can understand the status of the airplane in its entirety and optimize it in real-time (and, with a data connection to dispatchers and the air-traffic control system, soft-

1. We use lowercase *internet* to refer generically to a group of interconnected networks, and uppercase *Internet* to refer to the public Internet, which includes the World Wide Web.

ware can also understand the airplane's relationship to other planes and to the airspace around it).

The infrastructure of the Internet is highly flexible and scalable. Once a system of machines is brought together on a network, it's easy to add new types of software intelligence to the system, and to encompass more machines as the scope of optimization expands.

Software abstraction makes the physical world accessible

Web services mask their underlying complexity through software interfaces. Need to convert an address to latitude and longitude? Google's geocoder API² will make the conversion almost instantaneously, masking the complexity of the underlying process (text parsing, looking up possible matches in a database, choosing the best one). Geolocation thus becomes accessible to anyone building a Web site — no expertise in cartography needed. These services become modules in Web applications, which are designed with minimal assumptions about the services they use so that a change or failure in one module won't break the entire application.

In the same way, the industrial internet presents machines as services, accessible to any authorized application that's on the network. The scope of knowledge needed to contribute to a physical-world solution becomes smaller in the process.

Making a furnace more efficient, for instance, might involve some combination of refining its mechanical and thermal elements (machine design) and making it run in better relation to the building it's in and the occupants of that building (controls). The industrial internet makes it possible to approach these challenges separately: connect the furnace to a network and give it an API that guards against damaging commands, and the control problem becomes accessible to someone who knows something about software-driven optimization, but not much about furnaces.

In other words, the industrial internet makes the physical world accessible to anyone who can recast its problems in terms that software can handle: learning, analysis, system-wide optimization.

2. <https://developers.google.com/maps/documentation/geocoding/>

At the same time, this transfer of control to software can free machines to operate in the most efficient ways possible. Giving a furnace an advanced control system doesn't obviate the need for improvements to the furnace's mechanical design; a machine that anticipates being controlled effectively can itself be designed more efficiently.

Optimization above the level of a single machine

With machines connected in Internet-like ways, intelligence can live anywhere between an individual machine's controller and the universal network level, where data from thousands of machines converges. In a wind turbine, for instance, a local microcontroller adjusts each blade on every revolution. Networked together, a hundred turbines can be controlled by software that understands the context of each machine, adjusting every turbine individually to minimize its impact on nearby turbines.

Optimization becomes more effective as the size of the system being optimized grows, and the industrial internet can create systems that are limitless in scope. Upgrades to the American air-traffic control system, for example, will tie every airplane together into a single system that can be optimized at a nationwide level, anticipating a flight's arrival over a congested city long before it approaches. (The current system is essentially a patchwork of space controlled at the local and regional level.)

Software intelligence, which relies on collecting lots of data to build models, will become smarter and more granular as the scope of data collection increases. We see this already in the availability of traffic congestion data gathered by networked navigation systems and smartphone apps. The next step might be cloud-level software that gathers, analyzes, and re-broadcasts other machine data from networked cars — the state of headlights and windshield wipers to detect rain, for instance.

Optimization can go beyond a single kind of machine to take into account external market conditions. "Each silo has achieved its highest possible level of efficiency," says Alok Batra, the CTO and chief architect for GE Global Research.³ "If we don't break down silos, we can't generate more efficiency. Nothing operates in isolation anymore. If

3. Note: GE has sponsored this paper. See the acknowledgements section.

you operate a manufacturing plant, you need to know about wind and power supplies.”

Substitution of software for assets

The industrial internet will, as Astro Teller⁴, Captain of Moonshots at Google[x], suggests, “trade away physical complexity for control-system problems.” As machines deliver their work more efficiently, we’ll need fewer of them and the machines themselves will become simpler.

Consider, for instance, that California’s state-wide electricity demand stays below 30 gigawatts about 80% of the time. For about 20 hours every year, though, it surges past 47 gigawatts.⁵ Utilities must build out massive capacity that’s only used during peak hours a few days each summer.

Flattening out those peaks could dramatically reduce the capacity needed to reliably serve the state’s electricity needs, and that’s a control-system problem. An interconnected stack of software that extends all the way from power plants to light bulbs — parts of which are sometimes called the “smart grid” — could gather system-wide context as well as local preferences to gently control demand, dimming lights during peak hours and letting temperatures drift slightly in buildings whose owners accept a financial incentive in return for flexibility.

Substitution of software for labor

Given a high-volume stream of accurate machine data, software can learn very fast. And, by transmitting what it learns back into a network, it can accumulate knowledge from a broad range of experiences. While a senior pilot might have 10,000 to 20,000 hours of flying experience, a pilotless aircraft operating system might log hundreds of thousands of hours in just a year, with each of many planes transmitting anomalies back to a universal learning algorithm.

4. <http://astroteller.net/>

5. http://www.epa.gov/cleanenergy/documents/suca/ee_and_dr.pdf

U.S. manufacturing productivity grew by 69% in real terms between 1977 and 2011⁶, in part because machines automated many low-level human tasks. In health care, similar gains have been elusive: productivity grew by just 26% in real terms over the same period as spending nearly quadrupled (and productivity — economic output divided by number of employees — is itself an imperfect measure for what we want from our health care system).

The kind of automation that has revolutionized manufacturing has so far failed to revolutionize health care. Doctors and nurses spend much of their time reading machine data from sensors (everything from blood-pressure cuffs to MRI machines), matching patterns of symptoms to likely diagnoses, and prescribing medication within formal guidelines. As routinized as that work is, it still requires a great deal of human judgment and discretion that automation tools have so far not been able to provide.

The industrial internet will make the health care sector more efficient by providing intelligence on top of machine data. Software will ingest sensor readings and perform real-time analysis, freeing doctors and nurses to do work that requires more sophisticated and nuanced patient interaction. Progress is already well underway in home monitoring, which lets patients who just a few years ago would have needed constant monitoring in a hospital bed recover at home instead.

As automation did to factory workers, the industrial internet will undoubtedly obviate the need for certain types of jobs. If information is seamlessly captured from machines as well as people, we'll need fewer low-level data shepherds like medical transcriptionists (ironically, the demand for these types of jobs has increased with the introduction of electronic medical records, though that's largely due to the persistence of poor user interfaces and interoperability barriers). The industrial internet will automate certain repetitive jobs that have so far resisted automation because they require some degree of human judgment and spatial understanding — driving a truck, perhaps, or recognizing a marred paint job on an assembly line.

6. See http://www.bea.gov/industry/xls/GDPbyInd_VA_NAICS_1998-2011.xls and http://www.bea.gov/industry/xls/GDPbyInd_VA_NAICS_1947-1997.xls for output by industry and employment from 1998 to 2011; see http://www.bea.gov/industry/xls/GDPbyInd_FTPT_1948-1997.xls for employment before 1998. The health care statistics here refer to combined ambulatory, hospital, and residential care, NAICS codes 621, 622, and 623.

In fast-growing fields like health care, displaced workers might be absorbed into other low- or medium-skill roles, but in others, the economic tradeoffs will be similar to those in factory automation: higher productivity, lower prices for consumers, continued feasibility of manufacturing in high-cost countries like the United States — but also fewer jobs for people without high-demand technical skills.

Everything becomes a sensor

Any machine that registers state data can become a valuable sensor when it's connected to a network, regardless of whether it's built for the express purpose of logging data. A car's windshield-wiper switch, for example, can be a valuable human-actuated rain sensor if it's connected to the vehicle's internal network.

Software operating across several machines can draw from aggregate data conclusions that can't be drawn from local data. One car running its windshield wipers doesn't necessarily indicate rain, but a dozen cars running their windshield wipers in close proximity strongly suggests that it's raining.

Software operating across several types of machine data can also draw out useful systemic insights. Combined with steering-wheel, speed, GPS, and accelerator-pedal readings, a sensor-driven rain indication could warn a driver that he's moving too fast for road conditions, or help him improve his fuel economy by moderating his acceleration habits.

Machines built nightly

The Web brought about the end of the annual software release cycle.⁷ Provided as a loosely-coupled service on the Internet, software can be improved and updated constantly. The industrial internet will bring about a similar change in the physical world.

Some of the value of any machine is in its controls. By replacing controls regularly, or running them remotely and upgrading them every night like a Web service, machines can be constantly improved without any mechanical modifications. The industrial internet means that machines will no longer be constrained by the quality of their on-board intelligence. Development timelines for certain types of machines will

7. See <http://oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=4>.

become shorter as software development and hardware development can be separated to some degree.

Automakers, for instance, build cars with mechanical services that are designed to last more than 10 years in regular use. Entertainment and navigation systems are outdated within two years, though, and the software running on those systems might be obsolete in a few months. Automakers are experimenting with ways to decouple these systems from the cars they're installed in, perhaps by running entertainment and navigation software on the driver's phone. This scheme effectively gives the car's processor an upgrade every couple of years when the driver buys a new phone, and it gives the car new software every time the driver upgrades his apps.

It's easy to imagine something similar coming to the mechanical aspects of cars. A software update might include a better algorithm for setting fuel-air mixtures that would improve fuel economy. Initiatives like OpenXC⁸, a Ford program that gives Android developers access to drivetrain data, portend the coming of "plug and play intelligence," in which a driver not only stocks his car with music and maps through his phone, but also provides his own software and computational power for the car's drivetrain, updated as often as his phone. One driver might run software that adjusts the car's driving characteristics for better fuel economy, another for sportier performance. That sort of customization might bring about a wide consumer market in machine controls.

This could lead to the separation of markets in machines and in controls: buy a car from General Motors and buy the intelligent software to optimize it from Google. Manufacturers and software developers will need to think in terms of broad platforms to maximize the value of both their offerings.

Ultra-transparent markets replace regulation

The electricity market balances supply and demand on sub-second intervals, but data constraints prevent it from being truly transparent. As a result, efforts to reduce electricity demand (and its consequent impact on the environment) have typically been regulatory — mandating the phase-out of incandescent lightbulbs, for instance. Two el-

8. <http://openxcplatform.com>

ements are lacking: data-transmission infrastructure, which would send instantaneous price data from power producers to distributors, local utilities, and, ultimately, consumers; and some sort of intelligent decision making, which would take into account both instantaneous electricity prices and human preferences to decide, for instance, whether to run a dishwasher now or in 10 minutes.

The industrial internet promises to provide both data transmission and intelligent decision making, and in doing so it will create highly transparent, efficient, and comfortable markets down to the individual household level.

Security problems arise from systems that were built without connectivity in mind

Security vulnerabilities in the industrial internet often arise from the assumption that some system is isolated. Contraband connectivity invariably makes its way into any system, though. The best way to approach security is to assume connectivity and plan for it, not to avoid it entirely. Counterintuitively, Internet Protocol and other open, widespread internet technologies, by virtue of their having been under attack for decades, can be more secure than specialized, proprietary technologies.

Security issues are discussed in detail in the next section.

CHAPTER 2

Security

Adding software on top of machines and connecting them to networks creates tempting targets for malicious hackers. The evolution of industrial internet security is much like the evolution of PC security: many systems that are now being networked have historically enjoyed security by isolation, and, just as the original generation of PC operating systems didn't anticipate connections to the Internet, many industrial systems were not built with outside contact in mind.

The inherent scalability of software means that a single exploit can propagate fast; once discovered, an exploit can be used against lots of machines. Think of a car's odometer: the move to digital mileage counts, stored in software, makes it more difficult to tamper with the readout, but it expands the prospective target of an exploit from just one car (for mechanical odometers) to every car that uses the same software.

Tools like Shodan¹, a search engine for the Internet of Things, and Digital Bond's Basecamp², a database of industrial control exploits, illustrate the scale of the industrial internet and its vulnerabilities.

1. <http://www.shodanhq.com/>

2. <http://www.digitalbond.com/tools/basecamp/>