CITO Research
Advancing the craft of technology leadership

Six Essential Skills for Mastering the Internet of Connected Things

SPONSORED BY

neo4j

CONTENTS

Introduction

In the Internet of Things, practically any item from a smartphone to a carton of eggs becomes a node on a network. Information about that smartphone, smart car, laptop or tablet may be gathered from its interactions with WiFi networks and cell towers, while the comings and goings of that carton of eggs is gathered passively with barcode and RFID scanners.

This doesn't begin to capture the number of connections we're talking about. Users are connected to their smartphones (and the apps on those phones), devices are connected to each other (like smart meters), and health care devices such as MRI machines in a hospital may connect to an emergency dispatching center. Cisco's prediction that there will be 50 billion connected devices by 2020 is often cited in IoT discussions.[1] While CITO Research hasn't found data to quantify the number of resulting connections, two things are clear: there will be *vastly* more than 50 billion connections; and understanding and managing these connections is going to be at least as important as understanding and managing the devices themselves.

The number of connected devices and things is truly mind boggling, 50 billion by 2020 according to Cisco

With all this communication among devices, sensors, machines, and humans, the Internet of Things is really the Internet of *Connected* Things.

While the playing field for the Internet of Things (IoT) has been described many times, what is less well-known are the skills and technology needed to put those skills to work. In this paper, CITO Research takes a look at six essential skills you will need as the IoT moves off the drawing board and into production.

[1] "The Internet of Things" (infographic). http://share.cisco.com/internet-of-things.html

1. Envision Your Connected Product

The concept of *connected things* shifts the very meaning of "product," enabling you to consider not just what products are, but what they *could* become if connected in different ways. The product is no longer the thing, the device, but includes the halo of data, applications, and services that surround it.

A product can be anything. Any discrete application that's adding business value can be viewed as a product. It might be an application, a service, an API, or a "thing" in the conventional physical sense.

Making the most of your product's potential as a connected product requires design thinking that takes into account:

- The capabilities and characteristics of the device
- The data flowing to and from the device
- The applications able to access the device
- The users of the device
- Services that interact with the applications, are accessible to the users, and are used by the device itself
- Neighboring devices and the users connected to those devices

Imagination is key to unlocking the value of connected things

Imagination is key to unlocking the value of connected things. New products revolve around a new definition, one based on relationships and interdependencies, not just limited to the capabilities of an individual device. Thus, in a telecommunications or aviation network, the question "What cell tower is experiencing problems?" or "Which plane is going to arrive late?" becomes: "What is the impact of the problem on the rest of the network?" Understanding connections is the key to understanding dependencies and uncovering cascading impacts. These connections are key considerations for identifying opportunities for new services and products that leverage the IoT.

Recognizing the power of data and connections is vital to comprehending the potential value from all of the devices, items, and people that comprise the IoT. This requires moving away from a well-worn path: the idea of a product as a standalone entity is deeply entrenched. New possibilities can be unleashed by thinking of your product in a new, expansive, connected way.

2. Flexibility in Modeling Things

Another important consideration is choosing the right storage engines as the basis of your IoT application. Most IoT applications must work with dynamic and rapidly changing systems: new devices and applications are regularly coming online, each needing to fold seamlessly into the network. This requires a data model that can evolve without undue database and application re-engineering, and without impacting application availability.

Happily, a new generation of database technologies has emerged that does not require a predefined and fixed schema, a well-known limitation of SQL databases. Depending on your functional requirements, and on your speeds and feeds, options range from the likes of Splunk and Hadoop for offline analysis of high-volume data (such as sensor feeds) to key-value, column family, or document databases for real-time management of sensor data, to graph databases for managing connections. All of these technologies allow developers to create new fields or data elements very easily, with low impact to the application.

This sort of flexibility is pivotal for the IoT as new generations of devices with new types of data arrive in unpredictable ways. The NoSQL model, a general term for the set of technologies described above, easily accommodates changes and models the variation between types of devices, as well as the performance and scalability requirements that characterize IoT applications.

3. Richness in Modeling Connections

While product development is a high-level skill with lots of moving parts, it relies on having a detailed understanding of both the devices and their connections.

Graph databases are designed to easily model and navigate networks of data, with extremely high performance

The connections between devices and other entities can change faster than the data describing each thing. With telco data, each time you call a new person or authorize a new device, you make a new connection. The same is true in an industrial setting when a new piece of equipment comes online. It may look around for the relevant controllers or other devices that it needs to listen to or send data to. The powering up or down of a device may make or break dozens of connections. The natural way to represent such connections is using a graph.

In each of the instances of IoT data presented, relational databases, with their rigid structure, capture only a fraction of the story. Graph databases enable you to represent thousands of interconnections between nodes on a network and "slice and dice" the data based on any type of relationship you choose. Unlike relational databases, graph databases fit the NoSQL pattern described above, allowing flexible modeling of things. And unlike the other three types of NoSQL databases, graph databases are designed to easily model and navigate networks of data, with extremely high performance.

Graphs are a natural way to depict connections. Ask a telco engineer to depict a network on a whiteboard, and he or she will draw a series of circles representing nodes (cell towers), and a series of lines depicting connections (trunks). It turns out that most IoT applications require leveraging one or more data sets that are each highly connected in their own right, and often linked to one another (see Figure 1).

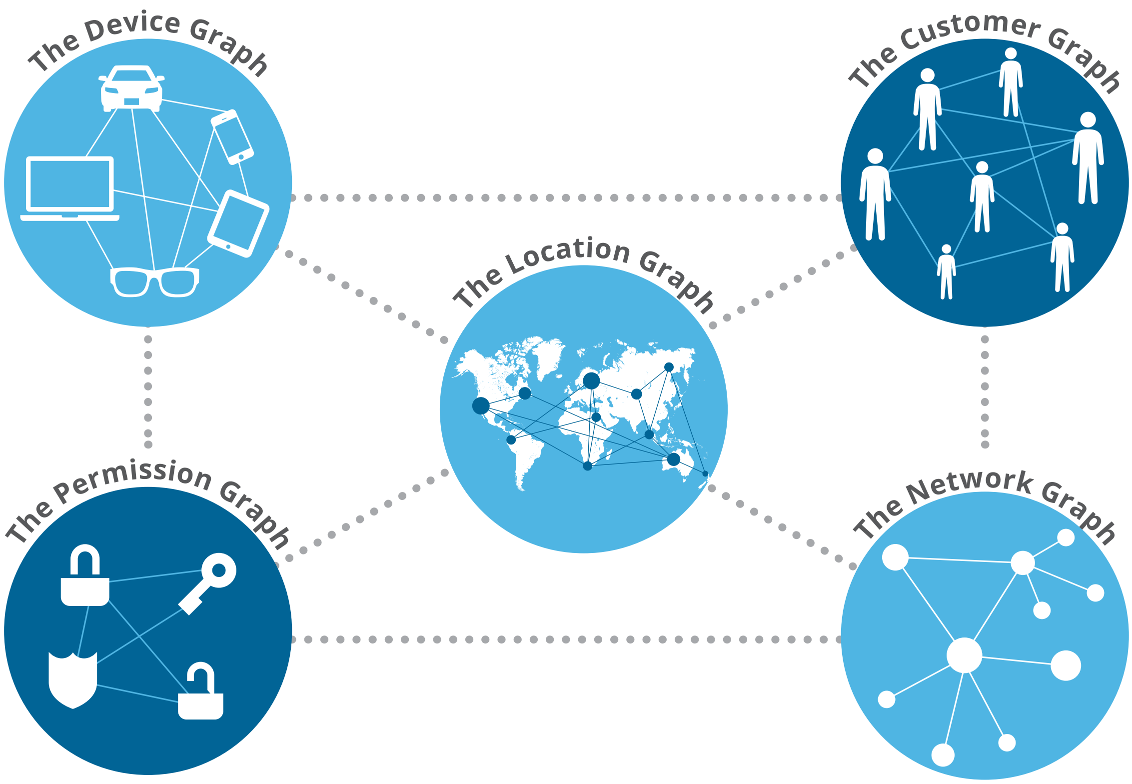

Figure 1. Connected graphs of the IoT

What is fascinating, and at times mind-numbingly complicated, about IoT applications is that all these graphs connect to each other. For example, the devices and users may be depicted on a map, bringing the device, customer, and location graphs into a single view.

Connections are more than lines between entities: they each include a richness of information, such as direction, type, quality, weight, and more—all of which can be represented, in a graph database, as an integral part of each relationship object. In this context, relationship attributes describe each connection. Attributes may indicate when the connection was created, the type of connection, the data related to the connection, and more. Just like the data describing a thing, the attributes of connections may change rapidly.

Graph databases process complex, multidimensional networks of connections very fast

While it is theoretically possible to represent a graph with attributes for both nodes and connections in any number of database management systems, it turns out that almost all production-quality applications based on fast-moving graphs use some form of a graph database. Graph databases are used because of the data model fit and for performance. Graph databases process complex, multidimensional networks of connections very fast. Companies like Twitter had to develop their own graph databases. Commercial forms of graph databases like Neo4j from Neo Technology have now become widely used for IoT applications.

Graph databases are essential to discovering, capturing and making sense of complex interdependencies, both for running IT organizations more effectively and to build the next generation of functionality for businesses. Whether a company seeks to optimize a network or application infrastructure, manage change or manage access for security purposes, the relationships are rarely linear or hierarchical. Rather, they form graphs—which are often quite dynamic. They change minute by minute (for example, cell tower uptime and downtime) or over time with personnel changes, mergers and acquisitions, and new applications being deployed and old ones retired.

Without a graph database, "I'd have to crawl all over the machines to figure out what they were doing, and then reassemble the pieces"

A graph database is a much better tool for capturing and modeling the interdependencies on a network to, for example, diagnose failures. A US-based software developer manages critical systems infrastructure for high availability by capturing how devices are interrelated across the infrastructure. As he describes, without a graph database, "I'd have to crawl all over the machines to figure out what they were doing, and then reassemble the pieces," whereas capturing bi-directional interdependencies in a graph database enables him to capture such information "easily and naturally without having to define a whole mess of linear relationships between each device." A graph captures the network as a whole.

A US-based software developer manages critical systems infrastructure for high availability by capturing how devices are interrelated across the infrastructure. As he describes, without a graph database, "I'd have to crawl all over the machines to figure out what they were doing, and then reassemble the pieces."

4. Conquering the Query

When most people think of queries, they think SQL. Graph queries as expressed inside a graph database have a different and much simpler structure.

While it's true that simple graph problems could be handled by a relational database, it's not a good fit, and here's why. Relational databases are built on tables. Relationships are recorded by storing data in the same row, or by using data in one field to look up data in another row or rows in another table.

While simple graph queries can be run in a relational database, albeit slowly, even mildly complex queries bog down and may not even return an answer. Such queries strain a data structure that was not designed to map connections. And while the incredibly complex and convoluted SQL may impress your co-workers, it's not generally sustainable, nor conducive to efficient development.

The simpler nature of graph queries enables instant scalability with easy-to-read queries. As one user who troubleshoots networks described, "[I] don't have to do a relational join between every machine with every other machine... 15 machines die and you don't know which one caused the problem—now imagine that's in a collection of 100 machines, or 1000 machines." A graph query can handle those relationships at scale, no matter how frequently they change.

But just as three generations of programmers mastered SQL, the current generation will have to understand how to write and use graph queries in their applications. This is an area of active innovation. Some graph databases are attempting to extend SQL. Neo Technology created a new query language called Neo4j's Cypher that is designed to efficiently query graph databases by describing a pattern and letting the database do the work of retrieving the desired pattern. Neo4j's Cypher is compact: it's not unusual for queries that are 50 lines of SQL to compress into just 4 lines of Neo4j's Cypher. eBay said their Neo4j solution is "...literally thousands of times faster than the prior MySQL solution, with queries that require 10-100 times less code."[2]

But just as three generations of programmers mastered SQL, the current generation will have to understand how to write and use graph queries in their applications

Skills in crafting and using the results of graph queries are crucial to gaining value from the IoT. It's important to choose a query language—and consequently an underlying database—that enables performance with ease and expressiveness.

[2] "eBay and Walmart Adopt Neo4j: The Graph Is Transforming Retail," Yahoo Finance, March 18, 2014, http://finance.yahoo.com/news/ebay-walmart-adopt-neo4j-graph-130300851.html

5. Capturing and Mastering the Data

The IoT will generate massive amounts of data because each sensor and device is continually capturing and transmitting measurements. Consider the volume of real-time data coming from the sensors in the IoT. These large volumes of detailed sensor data belong in a low cost big data repository, such as a key-value store or a column family database. Even then, many use cases won't justify keeping all of that streaming, real-time data for each and every sensor. Storing such data in Hadoop, or keeping it in log/file form and using a technology like Splunk to analyze the data offline, is another option for IoT applications not requiring real-time access to sensor data. (In our assessment, the offline scenario is likely more common than real-time streaming data analysis.)

Going back to the need for managing relationships, individual data points coming from the sensors don't typically need to be related to other individual data points coming out of other sensors. However, the relationships between the sensors themselves (and the rest of the network) do typically need to be managed. This naturally leads to a polyglot persistence model involving at least two data stores: if each sensor is related to other sensors in a graph database, that real-time data can be brought into the analysis on an as-needed basis from a (non-graph) bulk sensor data repository that stores the detailed tick-by-tick data from each device.

Graph databases for IoT applications couple well with different types of repositories for mass storage. Sometimes Hadoop works best, other times a column-family database like Cassandra or HBase fits the bill, and other times a key-value store like Redis can be used.

The limitations of relational databases in handling the flexibility and scale of IoT data has led to the marriage of graph databases and various big data repositories. Selecting the right combination of these two technologies for your application is another critical skill.

Graph databases for IoT applications couple well with different types of repositories for mass storage

6. Building Connected Applications

If you master the five skills described so far, the task then is to knit everything together to create the right applications. The product design represents the surface you want to provide to users. The graph databases used to model things and connections, the queries used to make that information useful, and the big data repositories to get the needed detail must all be put to work to provide value.

If you master the five skills described so far, the task then is to knit everything together to create the right applications

The challenge of IoT applications is the many different layers in most deployments, each of which can contain code and store data. Platforms like Zebra's Zatar provide a way to create an application in the cloud to supplement the capabilities of the device. ThingWorx, a PTC company, on the other hand, has a more flexible deployment model that allows code to be deployed in the cloud, on dedicated servers, and on any devices that can support its application stack. IoT applications are often themselves distributed like a graph, which requires a different way of thinking.

Here are some of the ways that IoT applications have come to life to create new products. Using graphs, a retail chain is able to see the waxing and waning of demand for products across geographies, and to reroute shipments to stores in which a particular item is in demand and low in stock. Manufacturers can use historic graph data to chart seasonal demand and project revenues based on that data for the next four quarters. The new products are freight optimization and demand forecasting.

An insurance provider can recognize the interrelationships and interdependencies between several seemingly unconnected people involved in a rash of car accidents. A deeper dive can reveal that they are involved in a complex insurance fraud ring involving hundreds of thousands of dollars in fraudulent claims for staged car accidents. The new product becomes advanced fraud detection.

A company can use the IoT to manage and secure its servers, specifying how they are secured, who has access (and uses that access), which servers are used, and with what frequency. Graph data may reveal that a fifth of servers are dormant and have fallen under the company's radar, thus exposing them to penetration. The new product becomes advanced system security.

Graphing the IoT at AmanziTel

AmanziTel is the leading provider of end-to-end customer experience management and service quality management solutions for telco providers. Their wireless networking products serve billions of mobile subscribers across several continents. AmanziTel guarantees the uptime and quality that telcos in turn guarantee their subscribers.

Graph data is key to AmanziTel's offerings. The company provides advanced intelligence for cellular networks, including statistics and measures of success defined by its telco customers. AmanziTel recognized two key needs in its migration to graph data—a *schema-less structure* and *performance.*

The schema-less structure is necessary as customer requirements change rapidly, and each customer requires different performance metrics and a flexible structure. Only graph databases can keep pace with those evolving requirements.

In terms of performance, graph databases can scale with increasing data volumes—and that data grows with each subscriber, each new cell tower, and all the traffic created by apps. Performance can only be supported by graph databases, not by rigid relational databases.

AmanziTel customers can capture signal strength in a graph, store and analyze that performance over time in a graph database, and instantly correct (reroute) based on location-based graph recommendations.

With no restrictions on data structures or on the data that is captured, a graph database can represent complex data and abstract concepts at the same time, all using the same database. The new product becomes guaranteed uptime (and cost avoidance in the forms of fines and denial-of-service claims).

In all the use cases described, the data is transient, grows with new users and activity, and creates ever-evolving products and opportunities based on connectivity and the performance of that connectivity.

Conclusion

The Internet of Things is really the Internet of Connected Things. The answer to the complexity and interconnectedness of the IoT is not an equally complex data structure; the answer is to reduce the wash of data to its common denominators. We can see that this requires not only one type of technology, but several, and that one very clear conclusion is that managing the connections, or relationships, present in IoT is a particularly big challenge. Graphs have evolved as a natural way to represent connected systems, and graph databases are a natural choice for managing the connected needs of IoT applications, just as Hadoop and NoSQL databases adapted to large volumes of log and sensor data clearly play a role in managing the vast volumes of sensor data often associated with IoT applications.

Graph databases will become the new norm for representing the connections inherent in the IoT

As we have seen, new skills are required to unleash the power of the IoT. Companies that have succeeded so far have moved aggressively to bring these skills inside their organizations or find partners who can provide them.

CITO Research believes that graph databases will become the new norm for representing the connections inherent in the IoT, embodied in the five graphs cited earlier: Device, Customer, Location, Permission, and Network graphs. Graph databases are already fast on their way to becoming the norm in telecommunications, perhaps the largest early adopter of the IoT. While telecommunications has been leading the charge, opportunities abound across all industries, and the discourse has already begun in industries ranging from retail to manufacturing to consumer electronics.

CITO Research predicts that within 5 years, IoT applications across all industries will outpace telco usage, as more and more industries recognize graph databases as the most effective tool for representing, analyzing, and monetizing IoT data.

The challenge for everyone in business today is to look at their business problems from a new angle. How will fresh thinking about your products as connected products change your business? What new opportunities does it enable? Does your definition of a product need to shift? Could your products become services? What data, and what relationships between data, would drive that opportunity? It will take creativity and vision to answer those questions for your business, but the time you invest in developing that vision could be game changing.

Learn more about graph databases and IoT ▸

This paper was created by CITO Research and sponsored by Neo4j

About Neo4j

Graphs are everywhere. From websites adding social capabilities to Telcos providing personalized customer services, to innovative bioinformatics research, organizations are adopting graph databases as the best way to model and query connected data. Neo4j researchers pioneered the first modern graph database back in 2000, and have been instrumental in bringing the power of the graph to numerous organizations worldwide, including more than 50 Global 2000 customers, such as Cisco, HP, Accenture, Telenor, eBay, and Walmart. Serving customers in production for over a decade, Neo4j is the world's leading graph database with the largest ecosystem of partners and tens of thousands of successful deployments.

Neo Technology, the company behind Neo4j, is a privately held company funded by Fidelity Growth Partners Europe, Sunstone Capital and Conor Venture Partners, and is headquartered in San Mateo, CA, with offices in Sweden, UK, Germany, France, New Zealand, and Malaysia. For more information, please visit www.neo4j.com.

CITO Research

CITO Research is a source of news, analysis, research and knowledge for CIOs, CTOs and other IT and business professionals. CITO Research engages in a dialogue with its audience to capture technology trends that are harvested, analyzed and communicated in a sophisticated way to help practitioners solve difficult business problems.

Visit us at http://www.citoresearch.com