

Case Study

Medical Equipment
Manufacturer

Custom Medical Equipment Manufacturer

Detailed Visibility into Complex Workflows**INDUSTRY**

Life Sciences

USE CASE

Process Improvement/
Real-Time
Recommendations

GOAL

Improve efficiency
and consistency of
manufacturing process

CHALLENGE

No visibility into status of
orders during complex
custom workflows

SOLUTION

Create a generic and
modular workflow engine
using Neo4j

RESULTS

- Instant order status for customers
- Fine-grained data views for managers, technicians and analysts
- Detailed visibility to drive process improvement

A successful medical equipment manufacturer required visibility into the myriad tailored workflows needed to produce customized products for patients. To support continued growth and improve operations, the manufacturer needed a workflow system that provides fine-grained data on its production processes to effectively scale, automate and innovate.

The Company

An established vertically integrated manufacturer of custom medical equipment leads its field through ongoing investment in research and innovation, enjoying year over year growth since its founding 50 years ago.

The Challenge

Custom medical products are highly personalized. A complex network of manufacturing processes is needed to satisfy patients' needs. A high volume of custom orders moving across multiple plants adds further complexity to the usual tasks involved in tracking orders from receiving to shipping.

As order volume increased and more systems were added on the plant floor, management's visibility into products moving through the custom manufacturing process diminished.

Technicians created numerous manual sub-processes to deal with customizations. They had to make routing decisions on the fly, often under pressure. Variability among technicians and processes led to quality control issues, but without visibility into these processes, there was no way to mitigate them.

Because so much of the custom manufacturing process is manual, even determining the status of an order is challenging. If a customer called, staff could say when the order was received, but could not update the customer about where their order was in the process or when it would be delivered.

Case Study

“Neo4j provides a natural representation of a workflow. The flexible data model provided by Neo4j, especially in comparison to a relational database, made it easier to accommodate our evolving needs. It enables us to efficiently traverse connected data – without complex queries and excessive JOINS – and make real-time decisions.”

– Senior Software Engineer

The Solution

The company's culture of innovation led their software team to look for new kinds of tools to turn their complex manufacturing process to their advantage. A graph database like Neo4j was a natural fit. The team began devising solutions to their workflow and routing problems using the connections in their data.

“We switched from a relational database to Neo4j for several reasons. The first is that Neo4j provides a natural representation of a workflow; after all, a workflow is a series of connected processes,” said a senior software engineer who worked on the project. “Next, our business needs are complex and continually evolving. The flexible data model provided by Neo4j, especially in comparison to a relational database, made it easier to accommodate these evolving needs. It also enables us to efficiently traverse connected data – without complex queries and excessive JOINS – and make real-time decisions.”

The team thought that creating a workflow engine was conceptually similar to a common graph database use case, [real-time recommendations](#). These systems use a person's data – such as their profile and system preferences – to make decisions about products to recommend. Similarly, a workflow engine takes a custom order – which includes information about the order, the customer, the customer's preferences and the technician working with the customer – and uses all of that information to recommend the most appropriate workflow and automatically route the order into that workflow.

Using Neo4j, the team developed a flexible workflow model. Their workflow engine adopted a polyglot persistence approach, using Neo4j for the live workflow and MongoDB for versioning and metadata.

The Results

The workflow engine built on Neo4j is generic, modular and reusable. It is used by different departments and accommodates both conditional workflow processes and failure steps. For even greater efficiency, some modules can be executed in parallel by different teams, accelerating order fulfillment.

Before implementing Neo4j, the manufacturer might miss a critical step such as a note to call the customer before creating the custom product. If someone missed such a notice, the product was made and shipped without checking with the customer. With the new workflow, a technician sees that there is a request to call the customer, so the order is then routed to the “Call Customers” workflow.

With fine-grained data into its workflows, everyone gains a better understanding of the state of a particular order. With this level of detail, managers see the product's path as it moves through the order process.

The workflow engine serves both external, consumer applications (like checking an order) and internal, employee applications (with specialized views for workflow designers, managers and technicians).

Neo4j is the leader in graph database technology. As the world's most widely deployed graph database, we help global brands – including [Comcast](#), [NASA](#), [UBS](#), and [Volvo Cars](#) – to reveal and predict how people, processes and systems are interrelated.

Using this relationships-first approach, applications built with Neo4j tackle connected data challenges such as [analytics and artificial intelligence](#), [fraud detection](#), [real-time recommendations](#), and [knowledge graphs](#). Find out more at [neo4j.com](#).

Questions about Neo4j?

Contact us across the globe:
info@neo4j.com
neo4j.com/contact-us