SIEMENS

Ingenuity for life

Internet of Things and Industry 4.0 predictions

The software and product development perspective

This paper outlines eight key predictions about the evolution of supporting technologies that will allow the Internet of Things and Industry 4.0 to become a reality. To make good software, and to mitigate its risks, a new generation of application and system development platforms will evolve from the more mature ALM tool platforms. Unified ALM will represent the fourth era in software development history.

A white paper issued by: Siemens PLM Software

www.siemens.com/polarion

Contents

Executive summary 3
Internet of Things 4
Industry 4.0 6
Good software is the key 8
"Good software" requirements 9
ALM predictions 10
Conclusions 13

Executive summary

The rumors are true: we have started living (and will soon be totally immersed) in the Internet of Things (IoT). We are entering upon a new Industrial Revolution that is defining new ways to create new kinds of products faster than ever before.

Close on the heels of IoT is another revolution, one spurred directly by it: Industry 4.0. It is like IoT applied to manufacturing, and it could easily swell into a new software crisis, as the expectations and requirements of future software coding, together with the explosion of software availability, could well exceed the ability to deliver good software on time.

Everyone will be a stakeholder in IoT, even if only as an end user. If you will be investing in this trend, or building or participating in IoT initiatives, or actively envisioning, designing or delivering innovation and solutions, your thoughts and concerns will gravitate towards software.

The future, including the part that is already here, is all about software. But not just any software. It's all about good software. Good software will need to be available, certified, compliant, flexible, re-usable, reliable, secure and safe. As a system developer, you may well be asking, "What's in this for me? What should I expect it to evolve into? What kind of technology will support me in the future?"

This paper outlines eight key predictions about the evolution of supporting technologies that will allow the IoT and Industry 4.0 to become a reality. To make good software, and to mitigate its risks, a new generation of application and system development platforms will evolve from the more mature ALM tool platforms. Unified ALM will represent the fourth era in software development history.

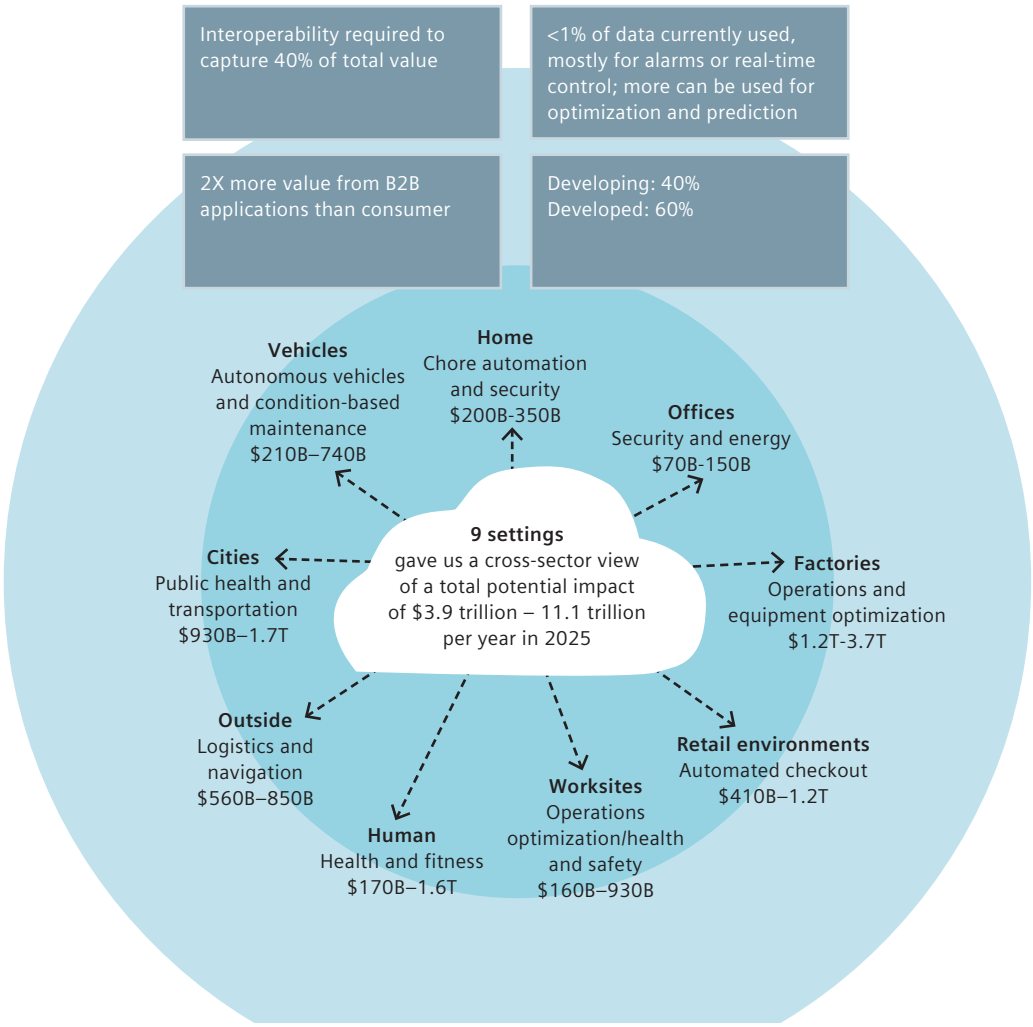
Internet of Things

The Internet of Things is a scenario in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT has evolved from the convergence of wireless technologies, micro-electro-mechanical systems (MEMS), and the Internet.

One could claim that IoT is a subset of the Internet of Everything (IoE). The other components of "Everything" are complex, but as a simplification, consider that "People" + "Things" = "Everything."

Regardless of how you define the details, *The Huffington Post*, in the infographic below from June, 2015, estimates the potential value of IoT in the range of $3.9 to $11.1 trillion per year by 2025, somewhere between the actual gross domestic product (GDP) of the U.K. and U.S. – certainly worthy of investment.

Where is the value potential of the Internet of Things?



From a user perspective, things will be connected, but on the technical side IoT will connect pieces of software code that will be installed everywhere. But software is, by definition, buggy. In 1996, it was estimated that in every 1 million lines of code there were 100,000 bugs (pretest), and best-in-class organizations were shipping such code with around 1,000 undiscovered bugs.[1] We are not doing any better in the third millennium. Due to software issues, Honda recalled 2.5 million vehicles in 2011, Toyota 1.9 million in 2014, and Chrysler 1.4 million in 2015. The U.S. Food and Drug Administration (FDA) warned about vulnerabilities in connected medical devices in 2015.[2]

It is exciting to think about the visionary IoT use cases that are foreseen and filling the news nowadays:

- Your car booking your massage while you take the road to the wellness center.
- Your refrigerator sending the list of goods you need at the grocery store and then setting your self-driving car to stop there while taking you home.
- Your doctor diminishing the amount of fat in the cream your coffee-machine normally puts in your cup every morning, responding to an alarm from your connected belly-button digital piercing.

On the other hand, we can think about the consequences that hidden software issues, badly designed systems or forgotten requirements can have on our fully automated future days. We can foresee spending hours trying to convince a concierge that we did not intend to book a hotel room for two years, or imagine finding 3,000 bottles of beer in front of the house because the refrigerator decided that last weekend's big party is a normal consumption pattern.

Scenarios can become nightmarish. Suppose the fitness app that came with the house won't unlock the front door unless we complete a workout by running for 30 minutes while wearing a tuxedo due to a glitch between the treadmill sensor and the wardrobe app. What if the 6-drop coffee turns into a two-liter monster mug thanks to a special promotion by the supplier? The list goes on, and the real scenarios that will make news will probably come as the unimagined surprise they always seem to be.

In the Internet of Threats (as pessimists tend to re-label IoT), the complexity of systems will explode. Apps installed in smartphones will normally interact with cloud software systems directly or indirectly connected to pieces of code (what we actually call embedded software) running on some smart device.

The IoT is all about software, but it needs to be all about good software. Good software won't result from just testing and retesting. It will come from a proper way to make software from the first glimmering of an idea to final disposal.

When selecting an investment, just as with choosing a new job, you will fare better to choose the right place. Simply pick a company making good software: a company that knows how to orchestrate their development processes by taking care of tracking complex system relationships; one that is able to trace from system requirements down to product components and software code, and is smart in orchestrating multiple test scenarios, and fast in coordinating complex product and software releases.

Industry 4.0

For some time now, various industry leaders whose ideas become more and more popular have considered that IoT applied to manufacturing production implies a Fourth Industrial Revolution called Industry 4.0.

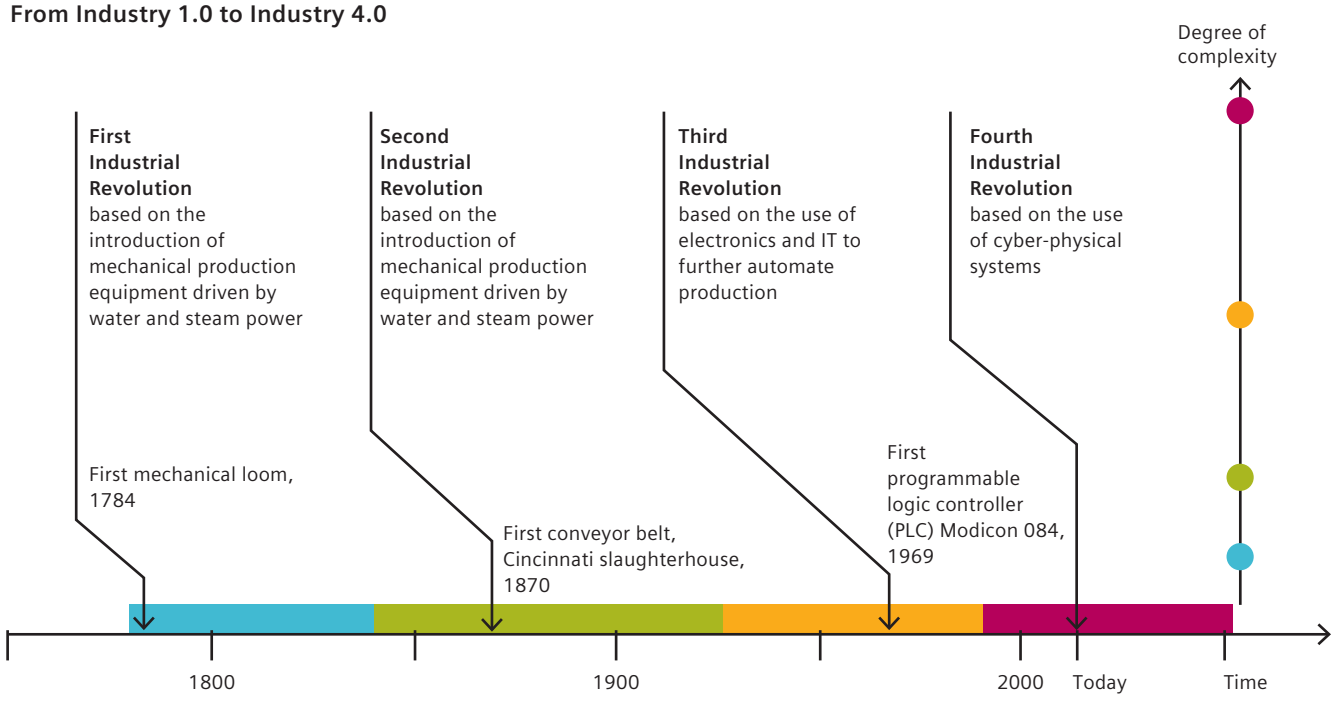
The first revolution happened with the introduction of the steam engine; the second is tied to the conveyor belt; and the third started when software, through PLCs, became part of products.

"Real and virtual worlds are literally colliding at the juncture of the Internet of Things, embedded software, social networking and autonomous decision-making.

The driving forces today that make Industry 4.0 viable include punctuated innovation, virtualization, shrinking globalization product lifecycles; and a burgeoning use of embedded software. The industrial segment is shifting toward digital manufacturing. This has been evidenced by 3D printing, which will eventually enable mass customization in the industrial world."

Carolyn Mathas
EDN

From Industry 1.0 to Industry 4.0



The new revolution starts at the time when more variability and innovation in products is coming from software than from hardware.

This has already happened with cellular telephones. Features in smartphones are updated and extended continuously through the various app stores, and people physically change their phones every one to two years. In the automotive industry, this shift hasn't happened yet. To get a new feature in your car, you most likely need a new car. But change is coming. For example, Mini is thinking of offering an on-demand and pay-when-you-need-it heating system in their car seats.

Professor Wolfgang Wahlster from the German Research Center for Artificial Intelligence (DFKI) clearly states that in Industry 3.0 we have "Machine + Software," while in Industry 4.0 we have "Software + Machine."

From an industrial perspective, the change is fairly consistent (and revolutionary): IoT will imply less inventories and more suppliers; fewer stock keeping units (SKUs) versus an enormous set of configuration options. We can easily imagine today that production machinery can be connected with its manufacturers so that repairing and downtime are limited and quite automated. We can also understand that all the logistics can get a boost with IoT, with automated deliveries upon requests coming from cyber-physical production devices and not from people.

What is definitely more difficult to foresee is the impact of components, the behavior of which changes according to where they are used or mounted. Components will talk to each other and adapt their behavior (and their features) to the context. An example is an electric motor that, when installed into a car's door, always turns at 20 rpm, but when put into a washing machine turns at five to 100 rpm. So we will have fewer "products," less hardware, more software, more configurations and more variants.

If we push ahead even further and consider 3D printing, we have a situation where there is practically no hardware, just a rough piece of plastic or steel plus software models and designs that can create from it a chair or teeth or a tap.

In a revolution, compared to an evolution, some dramatic shifts happen. While evolving, an organism changes, gets better, grows, adopts new habits and adapts its body to a new climate.

In a revolution only a few survive. In Industry 4.0 some key industrial players will become extinct like dinosaurs, and new players will occupy their space. In Industry 4.0 only those players that are able to quickly evolve enough to face the revolution will have a significant role tomorrow.

This fact seems to be pretty clear to the German government – so clear that in October of 2013, Chancellor Dr. Angela Merkel announced €500 million of funding to help Germany (and Europe) take the lead in the new industrial revolution (or at least, to help their companies evolve enough not to be obliterated from the industrial geography).

Facts and trends

While the Internet of Things represents an evolution in technology, it has such deep impact on manufacturing practices that it implies a revolution in Industry.

Looking a bit closer, we can already identify some facts in manufacturing that are clear signals of a new era:

- Software and communication technologies are part of the game. No longer the paradigm of "this is the input, this is the output, write a piece of code."
- Time is short. Actually shorter. Really, even shorter than that. How to compete with steel when your competitors compete with software features?
- Parts are also software. And more and more a valuable corporate asset, no longer an externally built commodity.
- Software is buggy. Releasing without defects is simply impossible.

Beside these facts, we perceive the following trends:

- Changing locations versus fixed. Manufacturers tend to move their plants frequently.
- Modular versus monolithic. Not one single organization but a network of smaller entities (note the Alphabet reorganization of Google in summer of 2015).
- Distributed versus hierarchical. Peer teams collaborating toward the same goal are emerging as an alternative to top-down problem decomposition.
- Wireless versus wired. Generally speaking, lightweight infrastructures to support mobility, modularity and distribution.

Good software is the key

In the context of the Internet of Things, and to make Industry 4.0 a reality, software plays the vital role. And it will be under the spotlight for a long time.

As a software development community, are we ready to face the challenges that are coming in the next few years? As a product development or systems engineering community, are we ready to deal with so much software?

There are serious concerns that IoT and Industry 4.0 can rely on any software. They will both require and must feed themselves with good software. So it won't be "it's all about software" – it will be "it's all about good software," and good systems engineering making good products that contain, expose or interact with good software. Some of the most influential people in the technology world are of the same opinion. For example, Vint Cerf, often called "the father of the internet," said in a recent article: "As more such appliances are run by software, people will be increasingly reliant on programmers' ability to write good code."[3]

There are many ways to define "good software." Probably the first one that comes to mind is the following:

Good software is the software that just does what is supposed to do.

Easy. To the point. But surely not sufficient to survive the revolution. There are some high-level requirements that need to be fulfilled in order to be "good," or at least good enough.

"Good software" requirements

A piece of code, to be considered good, will need to be:

A

Available
This simply means that "there must be software," in the right place, at the right time and at the right price. Software availability will be one of the most recurring reasons of project failure. If we think that the software crisis we had in the late 1960s is over, we probably underestimate the impact of IoT and Industry 4.0.

B

Certified
Once available, software needs to be verified. As software will be everywhere, an internal testing team or outsourced quality services won't be enough anymore. The number of certifications and certification authorities will grow.

C

Compliant
We will witness the explosion of software-to-software interactions. New standards will be needed every day, and new ways of assuring compliance to these new standards will be crucial.

D

Flexible
We won't be able to update all the software in all the things at the same time. New products will come out every day and will need to interact with existing products. Existing products will need to be equipped with software flexible enough to deal with the innovation constantly happening around it.

E

Re-usable
Fortunately, we won't need to rewrite the full code every time, for every product, for every version, for every product variant. Obviously not, if we will be good enough at making it re-usable.

F

Reliable
Software will not just need to do what it is supposed to do: it will need to do it anytime, anywhere and in any context. More and more of our core assets will rely on software. The money is already there, but what about health, safety, property?

G

Secure
A stranger getting into your car while you are watching a movie in a cinema needs to get close to your car, break a window, etc. Once in the IoT, your car is accessible from all over the world: anybody can theoretically get into it from anywhere. Ensuring software security is another complexity elevator for IoT and Industry 4.0 software.

H

Safe
A defect in your mobile phone that breaks a call is normally less harmful than your car driving into the street window of a café at dinner time. So the world that is in IoT and also the world that is not in IoT need to be protected by the software that we put into things.

ALM predictions

In order to fulfill the requirements listed in the previous section, software developers, development teams and organizations will need to be supported by processes and tools. As the discussion on processes is outside the scope of this paper, we will further investigate how software development platforms will evolve in the future.

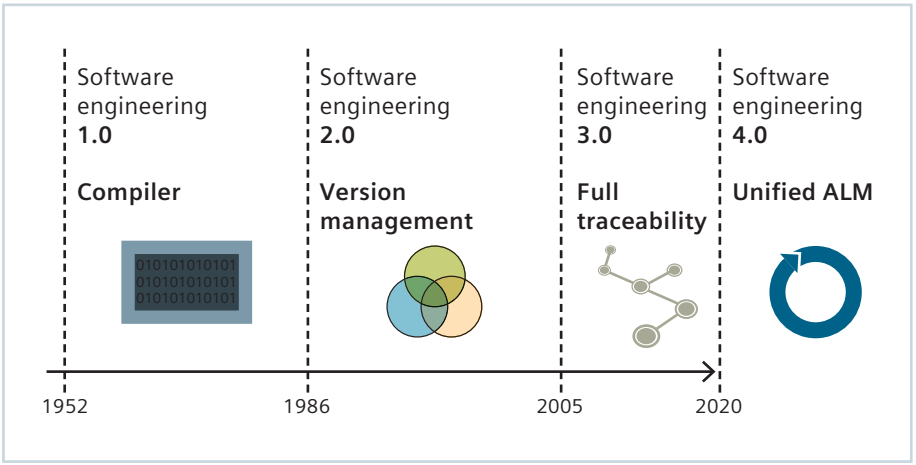
A quick look to the past shows that, in software engineering, or just in the way we have been making software, we have been supported by technologies that can be divided into three different evolutionary phases:

1. The compilers era
2. The versioning era
3. The traceability era

The last one is the current one, in which we use application lifecycle management (ALM) tools that are somehow connected to each other to support collaboration and information linkage and retrieval throughout the software or application development lifecycle. Sometimes we extend the scope to include DevOps, and sometimes we reduce the scope to application development lifecycle management (ADLM).

Either way, current ALM won't be enough to face the challenges of IoT and Industry 4.0. Current ALM does not do enough to assure us that we can fulfill the requirements of "good software." ALM needs to undergo an evolution (that for most vendors will really be a revolution) towards the concept of unified ALM.

In an ideal parallel with industrial revolutions, unified ALM will represent the fourth generation of the technology that the development community uses to make software, systems and products.



ALM-PLM predictions
Here are predictions about ALM-PLM evolution (or revolution), directly tied to the "good software" requirements we've already discussed, heading towards the next unified era:

Requirement: A, "Available" – it's all about collaboration. To deliver good software continuously and at the right time, all the barriers that prevent collaboration between all the project stakeholders must be eliminated and processes need to be automated.

#1
To allow the best collaboration, ALM-PLM platforms will cover more and more disciplines.

ALM-PLM interoperability will extend its reach to disciplines that are now at its boundaries and beyond. Model in the loop, release management, software and hardware in the loop (SIL, HIL) testing, and flash over the air (FOTA) are just a few examples. More will come.

#2
Processes will be simpler and their management will be handled by ALM-PLM interoperability versus people.

Whether we will go Agile, Enterprise Agile, Agile 2.0, Waterfall 3.0, or even MyOwnWay, the process knowledge will be stored and mapped in the ALM-PLM environment. Processes and approaches that cannot be fully automated will have a short life. Simplicity will arise by automation, not necessarily from the reduction of phases and practices.

Requirements: B, "Certified" and C, "Compliant": it's all about traceability.

#3
ALM-PLM interoperability will provide unified data navigation, unified user experience, unified point of access.

The extension of the reach of ALM-PLM interoperability, the amount of data to be managed and the diversity of the concepts will require a more stable and standardized user interaction.

#4
ALM will have tight interoperability links with product lifecycle management (PLM).

ALM and PLM need each other. We've been discussing this in other papers and we have also envisioned different levels of ALM-PLM interoperability. IoT and Industry 4.0 will increasingly need unified management of the software and hardware parts.

Requirements: D, "Flexible" and E, "Re-usable": it's all about configurations and variants.

#5
Version and configuration management won't be enough.

Mass customization and the explosion of different software-enabled features won't be supportable any longer by traditional version, configuration and change management approaches alone.

#6
Live branching, easy merging and variants management will emerge and lead.

As a consequence of prediction #5, we will see new ways of managing configurations: we will branch and merge all the lifecycle data, not just the software code. Certainly the variant management approach coming from product line engineering will be extended to software code and to application development artifacts, from requirements to test cases and beyond.

Requirements: F, "Reliable," G, "Secure," and H, "Safe": it's all about quality.

#7
Complexity in QA will increase exponentially: more tools, more test fields, more test cases.

Testing of devices, of systems and of systems of systems will require specific tools and approaches. We are already talking about "big testing data" and "testing big data": in a world where everything can be connected to everything, test scenarios will need to be managed in a fully automated and controlled way.

#8
Being able to seamlessly orchestrate disparate QA processes and tools and the ability to collect, analyze display testing results will be crucial.

As a consequence of prediction #7, ALM will need to contain, integrate and orchestrate a large variety of software testing tools, firing executions and then collecting and rationalizing test results to be displayed, compared, analyzed and connected to defects, software requirements, releases and so on.

8 ALM predictions for IoT & Industry 4.0

ALM-PLM will, in 5 years:

#1 *Cover more disciplines*
#2 *Simplify processes by embedding process knowledge*
#3 *Be unified*
#4 *Establish ALM-PLM interoperability*
#5 *Offer more than just versioning and configuration management*
#6 *Provide easy merging, branching and variants management*
#7 *Face a growing complexity in quality assurance*
#8 *Integrate disparate testing tools and results*

Timeframe
When will these predictions come true? As the need is real today and the gap to be covered is clear, we can expect most mature ALM vendors to be able to evolve into Unified ALM solutions with integration to PLM in the next five to eight years.

Conclusions

The quick evolution of ALM technologies, already a decade in progress, and boosted even more by wider adoption, is one of the key components that will mitigate the risk of the eruption of a new, more challenging software crisis in the next few years, as implied by IoT and Industry 4.0.

The software, product and system development communities will first concentrate on making their software toolchains (ALM, PLM) better than before. We will reshape our code handcrafting companions into a more mature, more perfect, usable and smart platform: Unified ALM in five to eight years to help us in delivering certified, flexible and reliable software on time and on budget.

Whether you make things to be connected in the internet (IoT), or you face the need of evolving your production (Industry 4.0), there are some challenges you must be considering now. Can you really afford to wait to have your ALM platform ready?

At Siemens PLM Software, with Teamcenter for PLM and Polarion for ALM we already have a proven, stable and unified ALM and PLM platform, in use by leading innovators since 2005. We are deeply involved in the evolution of lifecycle management platforms toward the needs of IoT and Industry 4.0.

References

1. *Computer*, March 1996

2. *Popular Science*, August 6, 2015

3. *PC World*, August 25, 2015 http://www.pcworld.com/article/2975738/internet-of-things/vint-cerf-sometimes-im-terrified-by-the-iot.html

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

AsiaPacific
Suites 43014302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

www.siemens.com/plm

55652-A3 8/16 F