Siemens Digital Industries Software

Make or buy? Using off-the-shelf vs. in-house tools for electrical and electronic systems development

Executive summary

Many modern products, including automobiles, aircraft and industrial machinery, contain sophisticated electrical and electronic (E/E) systems. E/E systems now deliver much of the product's functionality, but impact key parameters such as reliability, cost, and weight. While E/E systems development has traditionally been supported by in-house tools, powerful commercial-off-the-shelf (COTS) software solutions are now available to support E/E systems development, including wire harness manufacturing. Companies must weigh several factors when deciding whether to continue investing in in-house (proprietary) software development or to adopt COTS solutions. In particular, needs for flexibility, enterprise integration and IP protection impact this 'make or buy' decision in the E/E domain.

Nick Smith, Siemens Digital Industries Software

E/E systems development

Modern products including cars, aircraft, and complex machines such as medical imaging equipment have substantial electrical and electronic (E/E) systems that implement critical functionality. E/E system complexity has reached extreme levels, with parameters such as the quantity of embedded software code, interconnection count, and configuration variability growing to unprecedented levels. Multiple engineering domains fall within E/E systems development including:

- System behavior modelling
- E/E architecture design
- Software design, implementation, and verification
- Electronics design
- Data communications network design and verification
- Electrical system design
- Wire harness design and engineering
- Documentation and diagnostics provision

Natural adjacencies include requirements engineering, product and application lifecycle management (PLM/ALM), and 3D mechanical CAD (MCAD).

E/E systems development is a demanding and costly task. Errors such as sneak circuits, software bugs, or manufacturing mis-builds can have catastrophic consequences that compromise product safety and brand image, and incur significant financial impacts. To overcome these challenges, many companies are adopting formal model-based engineering methods[1]. It is now almost inconceivable that the design, optimization, verification, documentation, and logistical complexities of this domain can be addressed without the support of powerful engineering tools.

The three approaches to E/E engineering software
There are three basic approaches to providing software tool support to engineers:

1. **Non-specialized tools.** In the past, it may have been possible to use general purpose commercially available software such as Microsoft® Visio® or Autodesk® AutoCAD®. Such products are low-cost and enjoy a high degree of familiarity with most users. But the demands of the modern E/E domain, coupled with requirements such as data re-purposing, make such tools inefficient and ineffective for all but the simplest situations.
2. **Proprietary, in-house software.** Some organizations decide to develop their own tools, either directly or via a contracted organization, that are not made available to the open market. Sometimes these are implemented as extensions to general purpose tools, for example by writing complex Microsoft® Excel® macros.
3. **Specialized, COTS tools developed by software vendors.** Spurred by the growing demands of E/E systems development, a number of vendors now offer commercially available software solutions with varying levels of sophistication and scope. These standard software packages may be supplemented by extension (customization) capabilities.

This paper examines the trade-off between approaches two and three, a task often referred to as the 'make or buy' decision, specifically for the E/E systems development domain.

Make or buy?

'Make or buy' analysis is a common business situation with guidance available from numerous sources [2, 3, 4]. Factors relevant to the decision usually include (figure 1):

1. **Economics:** Companies will assess the total cost of ownership of COTS and proprietary solutions, including costs related to development and maintenance, switching costs such as user re-training, and opportunity costs. Cashflow and capital versus operational costs can also influence the decision. Overall, economic analysis focuses on quantitative assessment where economies of scale can be major factors. Opportunity costs, such as alternative uses of human or financial resources, are often overlooked and hard to assess.
2. **Intellectual property (IP):** E/E systems engineering solutions must protect proprietary knowledge, information or assets such as rule decks, manufacturing processes, and computer algorithms.
3. **Capability:** Can the software deliver what is needed, whether developed internally or via a contracted organization or COTS supplier? For software tools, this not only encompasses functionality, scalability, & user experience, but also compliance with specific IT demands such as operating system support, security policies and data protection such as GDPR . The assessment of software capability involves both technical delivery and the longevity of expertise and knowledge developed with the tool. Indeed, an over-reliance on key staff who may be approaching retirement is a common driver of the in-house to COTS transition.
4. **Strategy:** The core business philosophy and the degree to which vertical integration is desired will influence the 'make or buy' decision as well. Many businesses try to identify and reinforce their core competencies[7] to avoid diluting their focus. In the context of software tools, consideration must be given to software technology, not just the application domain. Such software technologies include secure software development processes [8, 9,10] background indexing[11], RESTful services[12], or HTML5/AJAX[13] support. In sum, companies must decide the extent to which software application development is a core competency, especially given the advanced software technologies available in today's COTS solutions.

   In addition to core competencies, companies must consider business flexibility. In-house software development personnel are a relatively inflexible cost that can be both expensive and personally traumatic to eliminate in the event of a business downturn. Sourcing software tools from third-party COTS or contracted vendors is far more flexible, especially if the vendor offers short term license rental options.

   Last, many COTS vendors provide their tools to universities and colleges at no or minimal cost. This makes it easier for companies to recruit engineers already familiar with the tools, which is not possible if in house tools are used exclusively.
5. **Intellectual Input:** The available breadth of intellectual input is an oft-overlooked factor when making a 'make or buy' decision for technology intensive products.

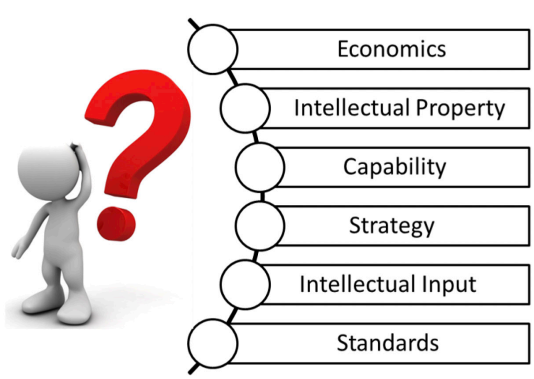

Figure 1: Factors impacting the 'make or buy' decision

Companies that favour in-house development risk becoming myopic to innovation, and thus progressively less competitive. Worse, this decline may be imperceptible from the inside. In contrast, COTS vendors can absorb and leverage a broad range of thinking, provided they have sufficient market reach and relationships. And because COTS suppliers have to compete commercially, they are driven to innovate and constantly increase the value delivered by their products.

This tendency towards innovation is related to, but distinct from point two above. Intellectual property may progressively migrate to become standard industry knowledge simply because a broad community reaches the same conclusion. This author's experience is that the extent of such migration is usually underestimated as multiple organizations often have the same 'intellectual property'. But, the dynamics of this process place a clear duty of confidentiality onto COTS suppliers not to leak genuine intellectual property to the broader market.

6. **Standards:** Industry standards encourage economies of scale and common practices. The E/E domain employs various standards such as the ReqIF[16] requirements interchange format, SysML[17] and VHDL-AMS[18] modelling languages, ARINC[19] and Ethernet[20] databus standards, AUTOSAR[21] embedded software standard, and FMI[22] dynamic model container. COTS software vendors are able to invest more time and resources to ensure their software complies with and reinforces these standards.
7. **Human attachment:** By this we mean the natural sense of pride and protection that employs feel towards something they have created. In this case, the individuals concerned may be the visionaries, coders, or maintenance staff associated with existing in-house software. This is often a powerful factor that can significantly cloud or distort a decision to transition to COTS software. It is a sensitive, but important management task to recognize when human attachment is undermining a healthy decision.

Technical consideration of COTS tools in the E/E domain

While all the factors shown in figure one are relevant to a 'make or buy' decision regarding E/E engineering software, three technical aspects are particularly important when assessing the capabilities of COTS tools (relating to point three above).

Flexibility

First is flexibility. COTS tools need to be highly flexible and adaptable to the specific needs of each company, team and engineer. This is because of the lack of widely adopted standards for certain aspects of the E/E domain and because of the existence of many different design and process flows. This is particularly true for suppliers who must support multiple OEM customers, each of whom may use different data formats. This challenge is often cited as a reason for pursuing in-house development because roadmap control is easier. Other E/E domain examples where flexibility is required include:

- Workflow variation
- User management
- Design and manufacturing process variation
- Data import, export, and reporting
- Simulation, checking, and verification requirements
- Object naming
- Graphical styling

COTS tools must therefore support a high level of configurability. If out-of-the-box behavior can be adjusted without resorting to external customization, ongoing maintenance costs are minimized and software licenses can be efficiently used for different projects. If the desired behavior cannot be achieved through built-in configurability, external customization (extensibility) will be needed. In this situation it is critical that the tools provide a future-proof application programming interface (API). In this way external customizations (plugins) can be maintained at minimum or zero cost.

Examples of specialized behavior accomplished using configurability and extensibility are shown in figures two and three. Figure two shows the application of customized rule checks to a schematic design. Figure three shows two graphical representations of a wire harness automatically generated from the same dataset.

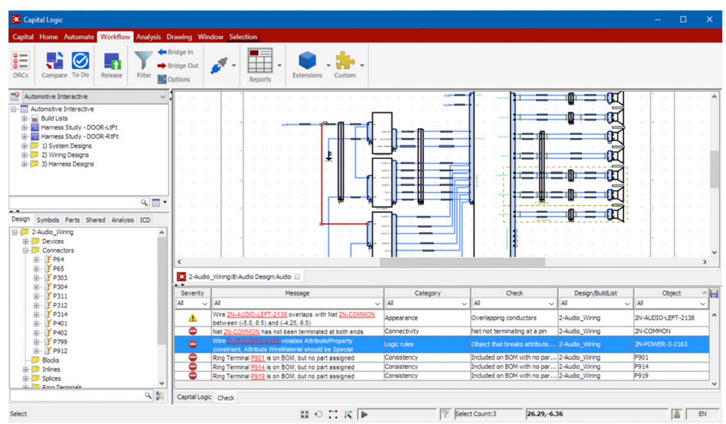

Figure 2: Use of customized design rule checks

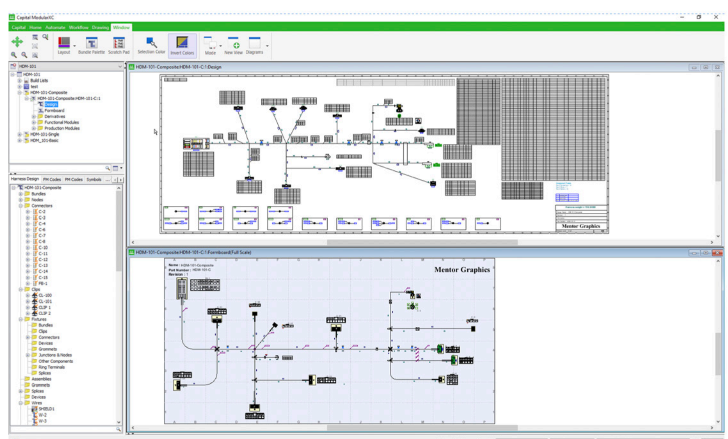

Figure 3: Alternative graphical representations of the same harness design dataset

Enterprise integration and the comprehensive digital twin

The next critical technical capability is enterprise integration, or the ability to link seamlessly with adjacent engineering and process environments. For E/E system development, the list of potential adjacent environments is remarkably long. It is therefore vital that COTS tools are designed to operate as part of an 'open ecosystem'. In addition to the more obvious adjacent engineering applications such as 3D mechanical CAD, common integration needs include PLM and ALM solutions, user authentication, workflow and release management, product planning, service documentation and diagnostics, manufacturing execution systems (MES), and factory equipment. Ensuring such enterprise integrations allows the E/E systems domain to contribute to a comprehensive digital twin of the product that models every product dimension.

The number of potential integration patterns is almost infinite. COTS tools must be architected to integrate with often unknown third-party environments as a core principle. The nature of integration architectures varies somewhat. Some adjacent environments are very pervasive, so it makes sense for COTS vendors to offer standard integration products, such as the leading 3D mechanical CAD platforms. But because of the vast array of enterprise integration patterns such tools must also provide a large number of integration hooks, such as the ability to publish and consume web services[24]. Just as with software extensions, it is important to consider which integration technologies to leverage to minimize or eliminate ongoing maintenance costs. Fortunately initiatives such as Open Services for Lifecycle Collaboration[25] and low code application development platforms[26] are making this easier.

IP and security

Third, protection of corporate know-how is frequently cited as a reason to reject COTS tools in favour of in-house development. This logic, however, prevents companies from gaining the benefits of economy, innovation, business focus and so on delivered by COTS solutions. A simple example of such IP could be 'we never allow more than six wires in an electrical splice because we have learned that more than six is unreliable'.

The solution is for COTS tools to include a mechanism by which IP can be captured privately and, just as important, systematically applied. Technically, this can be done by providing very rich configurability and extensibility capabilities. In this way, COTS tool behavior can be adapted to incorporate all manner of unique design and process IP, and to hide the IP from the outside world (including the COTS tool vendor). Examples are shown in figures four, five and six. Figure four shows part of a bespoke rule deck controlling wiring synthesis[27], in turn built from rule primitives like those shown in figure five.

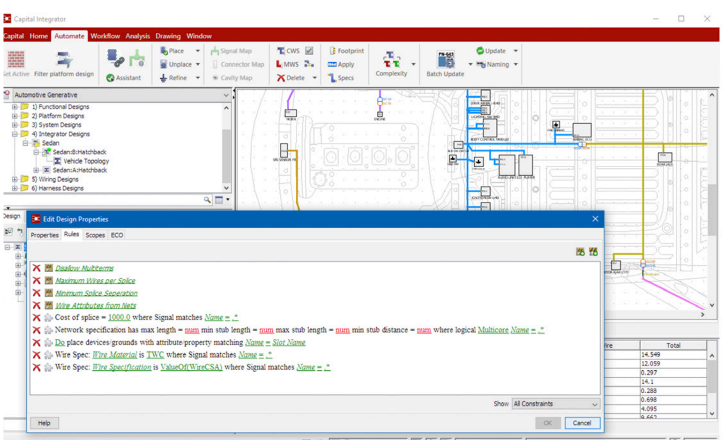

Figure 4: Snippet of a wiring synthesis rule deck

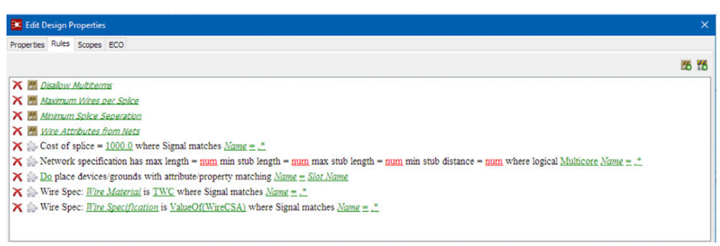

Figure 5: Examples of rule primitives

Figure six shows the dialog used to define bespoke manufacturing process patterns[28] against which wire harness manufacturing process trees can be generated. Similar technology can be used for detailed harness manufacturing cost calculation, which is an especially sensitive subject.

Interestingly, most major wire harness suppliers use COTS enterprise resource planning (ERP) and MES tools but, until recently, have developed their own proprietary tools for more specialized tasks such as manufacturing process tree calculation and optimization. This is because appropriate COTS tools with the characteristics described in this paper have only become available in recent years.

Simply the provision of infrastructure supporting the configurability and extensibility of COTS software is not sufficient. To support true proprietary protection, it must be possible for the customer, or their trusted third party contractor, to implement their own software configuration and extensions, i.e. to be independent of the COTS tool vendor. To support this, appropriate documentation and training must be readily available, and common programing languages such as Java[29] supported.

Of course, IP protection does not end with tool configurability and extensibility. Another aspect is the transfer of data between organizations, for example between OEM and supplier. It may be necessary to transfer proprietary information without exposing the proprietary content itself. For example, electrical engineers may need to transfer an electrical design with various proprietary simulation models that can be executed. The solution to situations like this is to encrypt the models so that only authorized staff possessing decryption keys can view their actual workings.

Engineering software must also be able to guard against malicious activity, such as the deliberate theft of IP. All tools and the environment within which they operate, whether COTS or in-house, should employ counter measures to detect and repair vulnerabilities. In this case, COTS vendors can provide superior capabilities because of their clear core competence in software engineering, including awareness of the latest security technologies such as secure communications between databases.

Finally, it is important to manage user access within an organization. Staff may be assigned to one project but restricted from another. For example, staff within an aerospace company may be assigned to either military or civilian projects, with access to the military projects tightly controlled. Modern COTS software solutions feature configurable privileges to allow only certain projects to be accessed, or indeed visible, by individual users or groups of users.

COTS software can deliver appropriate IP protection in each of these areas via a combination of configurability, extensibility, security technologies, and functionality.

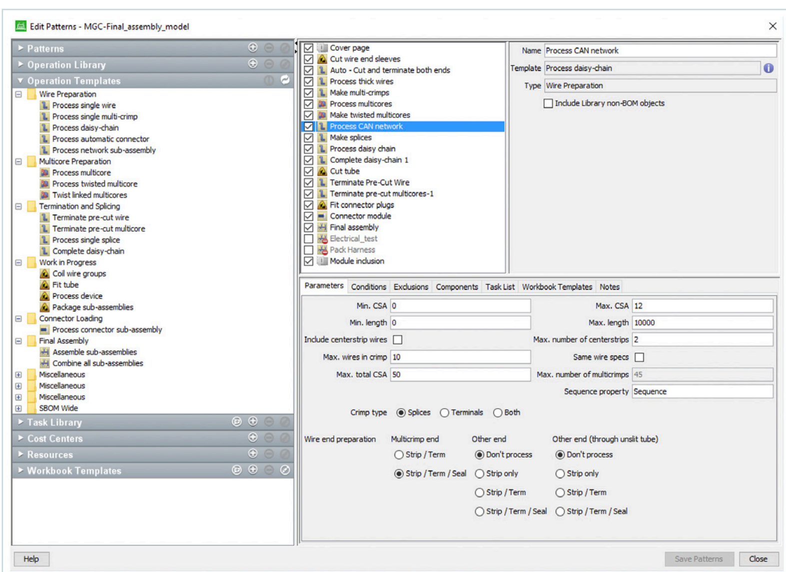

Figure 6: Dialog used to define wire harness manufacturing process patterns

Trends indicate COTS dominance

Many industries have faced the ‘make or buy’ decision. In the case of engineering software, the trend is clear: COTS suppliers have continually grown market share, diminishing in-house software and non-specialized tools almost to the point of extinction. Typically, a group of dominant COTS suppliers emerges, often complemented by an ecosystem of adjacent vendors. These large suppliers are able to achieve economies of scale as their revenues are sufficient to support significant development teams while remaining profitable. This means the COTS platforms become very powerful technically, making in-house development economically unattractive. At this point, factors such as innovation and strategic business focus also come into play, further diminishing the value of in-house development.

Other domains have already undergone the transition from in-house to COTS software solutions, including:

1. 3D mechanical CAD: Siemens Digital Industries Software, Dassault Systèmes, and PTC have become major COTS suppliers.
2. IC design: Mentor Graphics (now part of Siemens), Cadence, and Synopsys have become major COTS suppliers.
3. Enterprise resource planning: SAP, Microsoft, and Oracle have become major COTS suppliers.

In all three cases, in-house development can now only be justified for the most esoteric situations.

This pattern is clearly playing out in the E/E systems domain. In-house applications are being replaced by COTS tools and the virtuous circle of economies of scale leading to ever more effective COTS products is well underway. For example, all ten of the world’s top ten automotive OEMs now have COTS tools at the core of their electrical design environments. Tasks such as schematic capture are now very rarely best supported by in-house software, and subjects such as configuration control, cost calculation, and harness manufacturing engineering are rapidly moving in that direction.

A recent example in the E/E domain is data communications network design. This has often been accomplished using bespoke spreadsheet macros that are difficult to maintain and difficult to upgrade as new protocols emerge. These are then supplemented by hardware based testing that is difficult to make comprehensive for worst case scenarios. Sophisticated COTS network design and verification tools are now available that deliver rules based automation, timing analysis, and verification (figure 7).

Easy implementation of customizations and maintenance remains a strong requirement in the E/E systems domain as companies tailor processes to fit their specific needs. While IP capture & protection remains vital in some areas, especially harness manufacturing, this broad trend reflects both the tendency for good practices to migrate through industries and a sharper focus on core competencies such as engineering innovation.

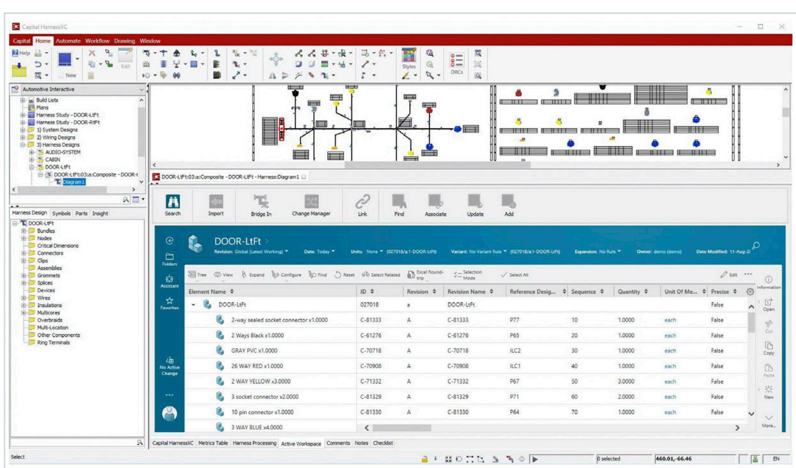

Figure 7: COTS data communications network design & verification tool

Conclusion

Several factors impact 'make or buy' analyses. All these factors apply to deciding between COTS and in-house E/E systems development software. Of these factors, perhaps the most difficult to manage is human: psychological attachment to previous endeavours.

In the E/E domain the decision now is usually to 'buy'. COTS tools are widely deployed, driven principally by their cost advantage and maturity, including mechanisms supporting easy customization and IP protection. These capabilities help companies to manage challenges such as security and the desire for sharper business focus. This trend matches the pattern observed in other domains. It is likely to accelerate as further economies of scale accrue, enabling COTS suppliers to deliver yet more powerful products. And as COTS tools proliferate, the cadre of familiar users and associated ecosystem will also grow, further reducing the attraction of proprietary software development.

Common Benefits Of COTS Software

- Improved strategic focus
- Improved sustainability
- Lower total cost of ownership
- Access to broad intellectual inputs
- Access to latest technologies

References

1. https://modelbasedengineering.com/faq/ 2015.
2. www.strategyand.pwc.com/media/file/Strategyand_Make-or-buy-sound-decision-making.pdf
3. www.atkearney.co.uk/documents/10192/317934/Make-vs-Buy-Revisited.pdf/71569ecf-c266-484b-91a8-3385885d5e4e
4. www.investopedia.com/terms/m/make-or-buy-decision.asp?layout=infini&v=5D&orig=1&adtest=5D
5. https://gdpr-info.eu/
6. www.mentor.com/products/electrical-design-software/resources/overview/knowledge-retention-strategies-for-auto-aero-companies-0b5b6729-a879-4935-b239-cac9ddc4f68c
7. www.managementstudyguide.com/core-competency-theory-of-strategy.htm
8. https://csrc.nist.gov/CSRC/media/Publications/white-paper/2019/06/07/mitigating-risk-of-software-vulnerabilities-with-ssdf/draft/documents/ssdf-for-mitigating-risk-of-software-vulns-draft.pdf
9. https://security.berkeley.edu/secure-coding-practice-guidelines
10. https://security.ucop.edu/policies/secure-software-development.html
11. www.lirmm.fr/mastodons/talks/Valduriez-Bigdata-indexing-2014.pdf
12. https://en.wikipedia.org/wiki/Representational_state_transfer
13. www.w3schools.com/ajax/default.asp
14. http://www.fastcompany.com/874798/why-innovation-matters
15. https://en.wikipedia.org/wiki/Requirements_Interchange_Format
17. https://sysml.org/
18. https://en.wikipedia.org/wiki/VHDL-AMS
19. https://en.wikipedia.org/wiki/ARINC_429#:~:text=ARINC%20429%20is%20a%20data%20transfer%20standard%20for%20aircraft%20avionics.&text=Data%20words%20are%2032%20bits,are%20monitoring%20the%20bus%20messages.
20. https://en.wikipedia.org/wiki/Ethernet
21. https://www.autosar.org/
22. https://fmi-standard.org/
24. https://en.wikipedia.org/wiki/Web_service
25. http://open-services.net/
26. https://en.wikipedia.org/wiki/Low-code_development_platform
27. www.mentor.com/products/electrical-design-software/resources/overview/what-is-wiring-synthesis--8a958fff-3d77-4d16-af63-e9c7bb909f33
28. www.mentor.com/products/electrical-design-software/resources/overview/wire-harness-manufacturing-process-management-49faca1f-24e8-483d-a964-85ed64150c8a
29. https://en.wikipedia.org/wiki/Java_(programming_language)

Siemens Digital Industries 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

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. The Xcelerator portfolio helps companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

82367-C2 9/20 A