MODULARIZATION

Modularization is the key to greater efficiency in machine engineering

The complexity of automation solutions is constantly growing. Manufacturers pass on the need for constantly changing production requirements to machine builders. This increases the risk of numerous individual solutions that are difficult to manage in engineering. To deal with this complexity, modularization comes into play. Combined with well-designed engineering software, it is the key to increase your engineering efficiency and flexibility.

siemens.com/efficient-engineering

SIEMENS

Modularization
Why modularization?

Modularization is a real game changer. If you have ever performed the same automation task several times with minor variations by creating each program variant individually, you will know that. And if you have ever copied and adapted program variants to save time, you will be aware that without variant management you will achieve exactly the opposite: a time-consuming, unmanageable variety of programs. It is much more efficient, assuming variant management, to retain the basic structure of the task and only exchange the specific variants on a modular basis. This saves time and resources and significantly reduces the potential for errors. It is also a major driver of standardization in the engineering industry.

The challenges
of modularization

However, the implementation of modularization varies greatly from project to project. There are many different reasons for this: For example, different team members may take different approaches. Or they may not see the added value of a modular project structure. Or they may be put off by the initial effort, which means a loss of efficiency before the benefits of modularization are realized. Or the problem arises much earlier because you don't know where and how to start with modularization and standardization.

We have therefore created a guide to implementation. It introduces you to modularization tools and their benefits, and shows you how to use them in projects.

The path to successful modularization
A modular and efficient engineering workflow requires the use of the following paradigms:

1. Rely on industry standards
2. Merge reusable modules
3. Perform consistent version control
4. Manage dependencies

1. Rely on industry standards

Choosing the right programming languages and adhering to established standards are the cornerstones of effective modularization. Programming languages provide a more abstract approach to problem solving. They allow engineers to focus on the design and logic behind the automation, rather than on hardware-related details. Different programming languages with adapted structures are suitable for different use cases. Established languages in automation are based on circuit diagrams. These can depict rigid processes and their dependencies well, but are limited in their degree of abstraction.

High-level languages from software development offer good standardized approaches. The IEC 61131 standard defines some languages that make elements of high-level languages accessible to automation, such as Structured Text. In addition, general industry standards such as PLCopen, ISA-88 or OMAC provide a common basis for development departments. This promotes interoperability and facilitates collaboration between different teams and companies.

2. Merge reusable modules

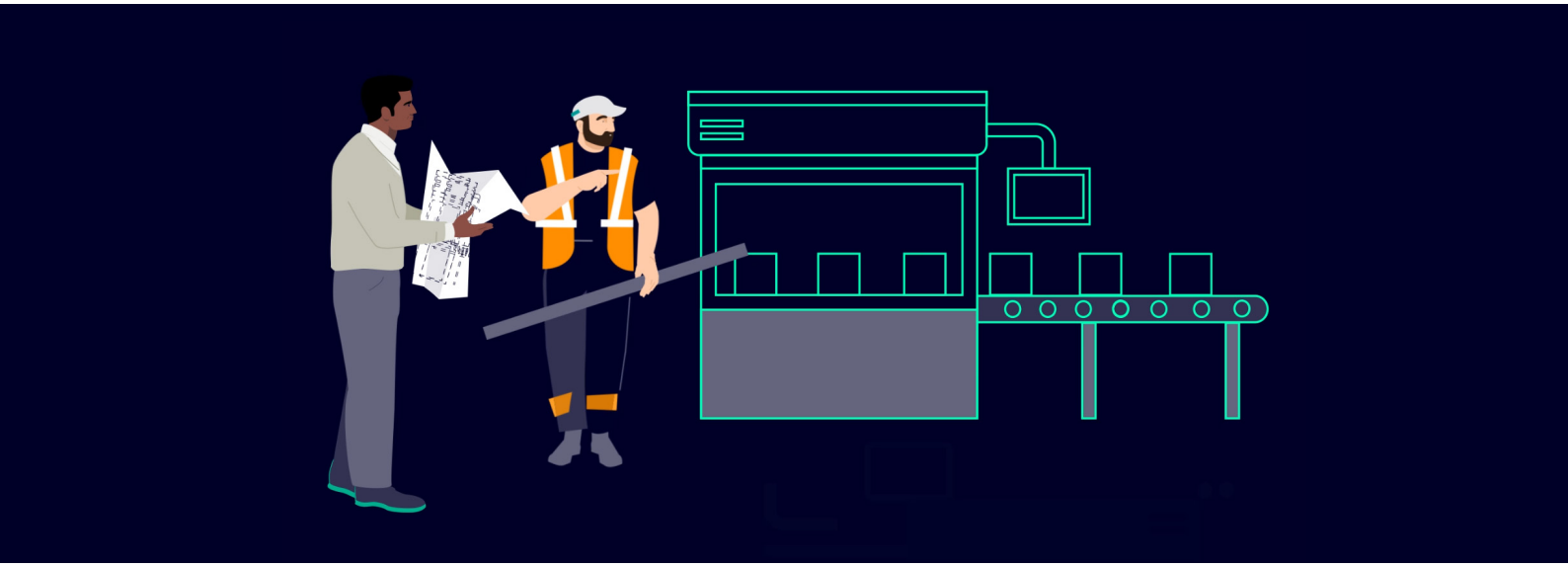
The use of containers is crucial to structuring code and information in a modular way. They store reusable modules. The modular structure allows engineers to reuse segments of code or data, reducing development time and ensuring consistency across projects. Containers can take the form of templates, libraries or packages.

Packages
Packages are collections of related functions, classes, or modules. Combined and bundled, they can be easily organized, managed, and used in a system.

Libraries
A library is a collection of pre-compiled code that provides specific functions or services that can be reused by other programs or projects. In the context of automation, for example, libraries provide functions for controlling machines, communication between different systems, or other automation tasks. The decisive advantage of libraries is their reusability. Instead of writing the same code over and over again, developers can refer to existing functions or modules in the library. This saves time, increases consistency, and reduces the likelihood of errors.

Optimize automation systems through the use of packages and libraries
In factory automation, packages and libraries enable modular and efficient development. They allow engineers to access predefined functions, modules, or components to implement automation tasks instead of developing each functionality from scratch. This reduces development time and improves the consistency and quality of the automation systems created. Packages and libraries therefore make a significant contribution to standardization in machine engineering.

3. Perform consistent version control



The places and systems that allow containers to be stored and managed are an important pillar of modularization in machine engineering. Version control interfaces such as Git or open-source communities provide platforms for organizing, storing and tracking changes to containers. They enable transparent version control and create an environment for collaborative development. This allows teams to work together efficiently without data loss or inconsistencies.

Git is a popular version control system. It manages changes to code and resources in software projects. It allows you to track changes, create different branches for parallel development, and consolidate changes into a central code repository. Repositories are databases or locations where Git stores project files. A repository contains the entire history of the project, including all changes, branches and versions. It can be stored either locally on a computer or in a remote location such as GitHub, GitLab or Bitbucket.

Often one-size-fits-all solutions end up being sub-optimal because their features and capabilities try to cover too much ground. But a simplified, all-in-one approach like TIA Portal is ideal for achieving more efficient engineering, especially when the engineering challenges within a plant or distributed industrial enterprise can span multiple disciplines, involve a myriad of development tools, components, and standards, and deal with all kinds of complexities. The following use cases illustrate how TIA Portal delivers more efficient automation engineering in a much simpler user environment.

Version control
Version control – or version management in general – is a system used to track, organize and manage changes to files or projects over time. It allows developers to document the progress of a project, track changes and, if necessary, revert to previous versions.

4. Dependency management

Library and package management tools are necessary to organize and manage containers efficiently. They allow engineers to quickly access the resources they need without wasting time searching for or manually managing modules.

One such tool is the package manager. It provides access to and automated management of pre-built modules of code and resources. Packages contain specific functionality, libraries or tools that are used in different automation projects. The package manager is the automation of automation, so to speak.

Summary
Mastering modularization

By using the technologies and methods presented, the barriers to the real challenge of modularization disappear. By implementing the abstraction levels, a simplified representation of the actual automation task is created in the sometimes laborious preliminary work. This is the key to modular engineering.

Siemens solutions such as TIA Portal or SIMATIC AX enable the efficient implementation of modularization. They provide engineers with platforms and tools to create a structured, modular environment for factory automation. They not only support efficient development, but also support code reuse and process standardization. This increases productivity and quality in automation.

Published by
Siemens AG
Digital Industries
Factory Automation
P.O. Box 48 48
90026 Nuremberg
Germany

For the U.S. published by
Siemens Industry Inc.
100 Technology Drive
Alpharetta, GA 30005
United States

Photos: Siemens AG, unless otherwise stated

Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.

© Siemens 2024