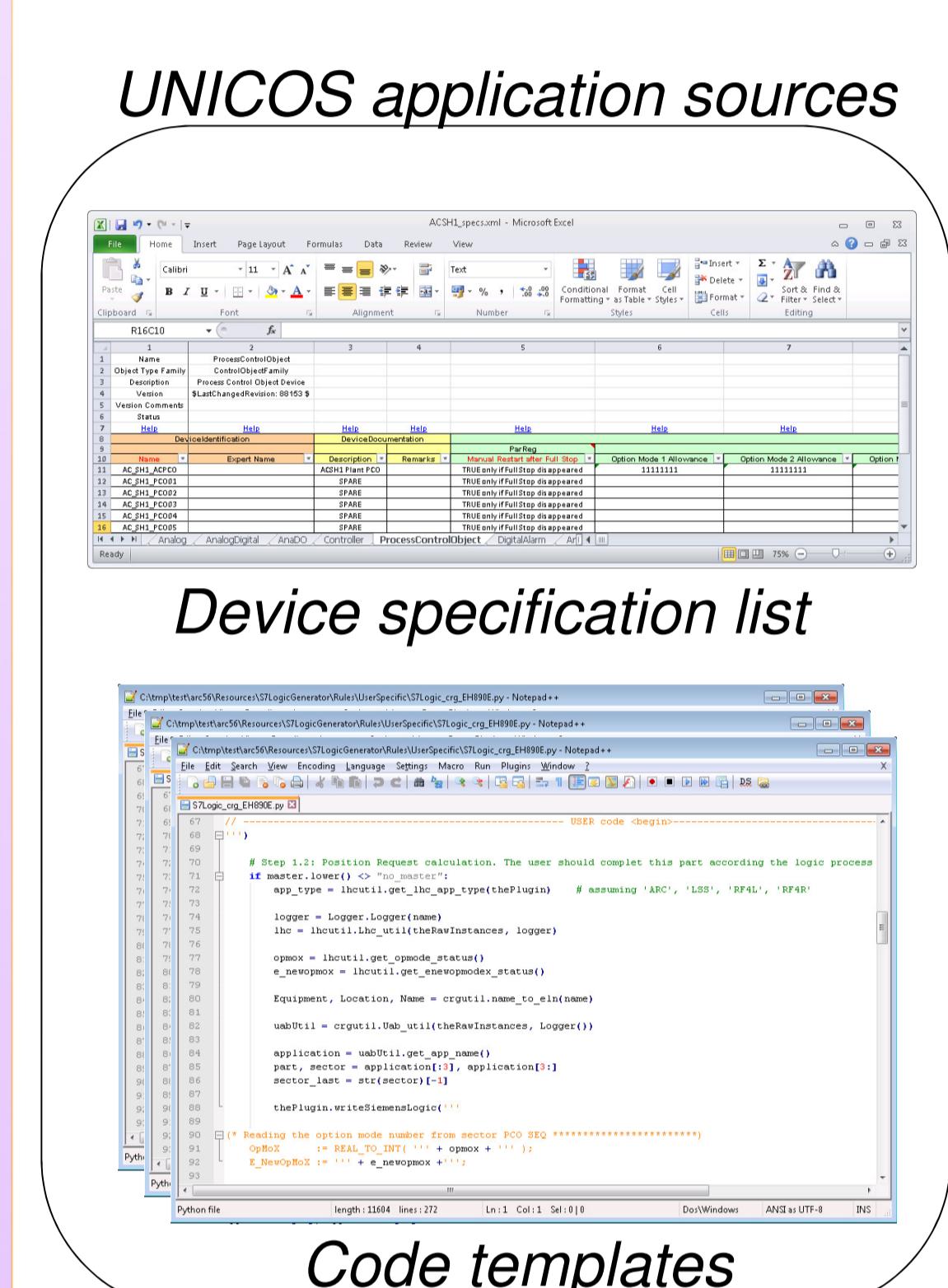


Abstract

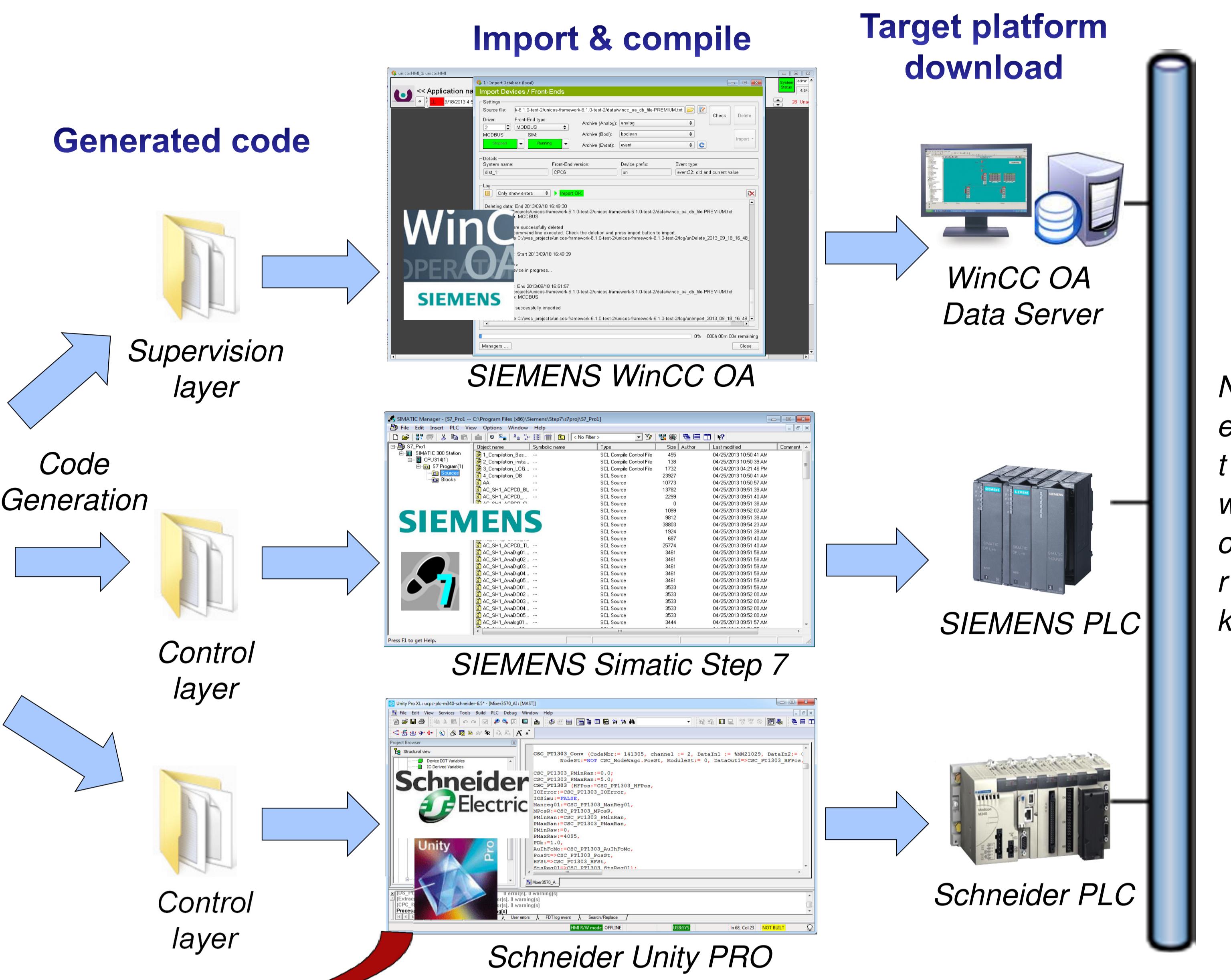
The UNICOS (UNified Industrial COntrol System) framework was created back in 1998 as a solution to build industry like control systems. The Continuous Process Control package (CPC) is a UNICOS component that provides a methodology and a set of tools to design and implement industrial control applications. UAB (UNICOS Application Builder) is the software factory used to develop UNICOS-CPC applications. The constant evolution of the CPC component brought the necessity of creating a new tool to validate the generated applications and to verify that the modifications introduced in the software tools do not create any undesirable effect on the existing control applications. The uab-maven-plugin is a plug-in for the Apache Maven build manager that can be used to trigger the generation of CPC applications and verify the consistency of the generated code. This plug-in can be integrated in continuous integration tools – like Hudson or Jenkins – to create jobs for constant monitoring of changes in the UAB framework that will trigger a new generation of all the applications located in the source code management.

Generation of UNICOS applications using graphical user interfaces

1 UAB provides a set of **wizards** with a user-friendly interface that are helpful during the **creation, configuration and development of UNICOS applications**, generating code for the **control and supervision layers** of the applications.

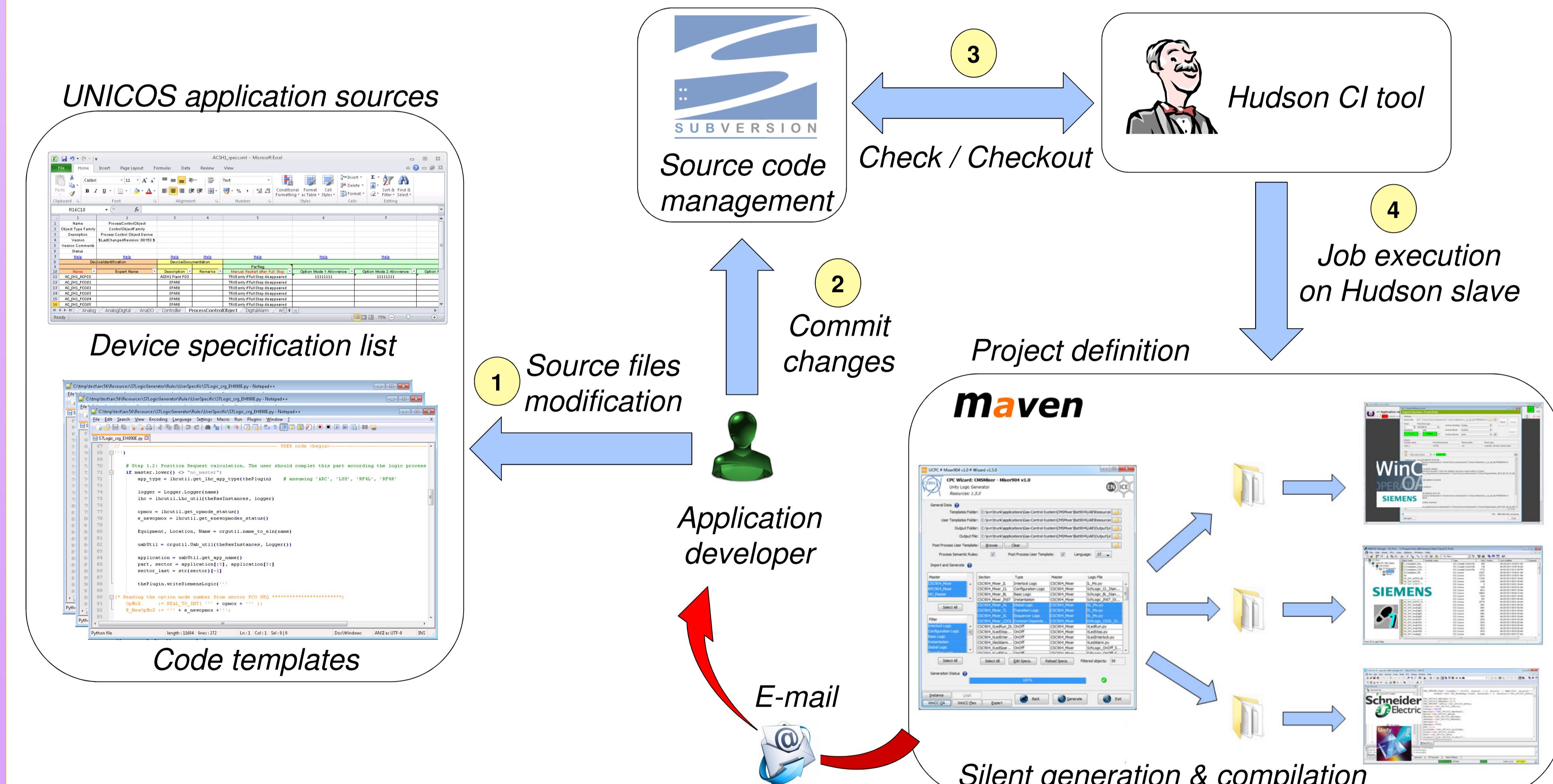


2 **Compilation errors** detected in the target platform software may require the **modification of the application sources** (device specification list or code templates), **regenerate the code** for the different target platforms and finally **import and compile** the generated code in the target platform software.



3 This **development workflow** is suitable to create new applications from scratch but is tedious when the developer is generating large or multiple applications. In these cases the generation procedure is a **repetitive, error-prone and time consuming task**.

Generation of UNICOS applications using continuous integration tools



4 UAB provides a plug-in for **Apache Maven** that can be used to **trigger silent generations** of UNICOS applications without further user intervention. **Hudson monitors changes** in the files located in the source code management tool and **triggers the code generation** of the related applications. If the generation is completed successfully, the generated code can be **automatically compiled** in the target platform software using the libraries provided by the vendors (SIEMENS, Schneider). If the generation **fails** at any point the user will be **notified by e-mail** of the failure.

The **uab-maven-plugin** is also useful during the **software development phase** of the different **UAB modules**. After a **source code refactoring** or the development of **new features**, it can be used to verify that the **generated outputs are equivalent** or the modifications did not introduce any **undesirable side effect**.

Conclusions

The use of continuous integration tools have been proven to be a best practice in software engineering during the software development and maintenance phases and assist with higher software quality. The UAB development team has made an effort to apply the same practices to the development of control system applications, providing tools to perform automatic executions of the code generators or plug-ins.

In the scope of UNICOS applications development, these tools allow to reduce the user interaction and avoid repetitive, error-prone and time consuming tasks. In the case of UAB software development, the tools provide a mechanism to verify the correctness of the modifications introduced in the software by comparing the generated outputs with previous versions.

These tools are currently being used when developing UNICOS-based control systems. The LHC GCS (Large Hadron Collider, Gas Control System), LHC Cryogenics and many HVAC (Heating, Ventilation and Air Conditioning) control applications are some examples of them. In summary, there are more than 50 applications using these tools which generate more than 14 million lines of PLC code.