ZDMP Blog

Fast track from computational models to computational software service

By Amirreza Baghbanpourasl, Daniela Kirchberger, Kastor Felsner, Profactor GmbH

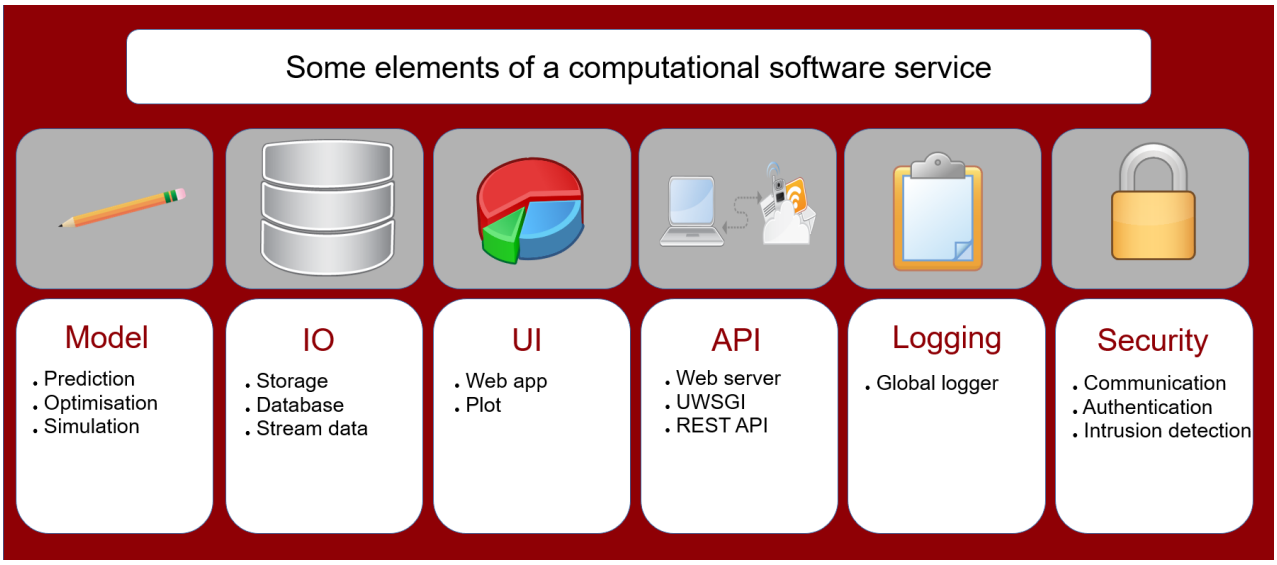
Some questions for you

- What are the examples of computational applications that can aid zero defect manufacturing?
- Do you know what elements a computational software service is consisted of?
- Is there any component in ZDMP that facilitates transforming a computational model to a computational software service?
- What are the applications of this component?

Computational software service and its elements

Digitalization of production has enabled a plethora of possibilities for taking advantage of computational techniques for controlling the production process [1, 2]. These techniques help to achieve zero defect manufacturing by providing applications, that can perform operations such as optimisation, prediction, anomaly detection, diagnostics, and simulation. What is common in all these applications is a computational core that performs a numerical or combinatorial calculation.

In such digitalisation efforts, a model (that is generally any mathematical or numerical function) is designed by a scientist/engineer. After creating a model that provides satisfying results, it is prepared for production. At this stage, an application is developed around the model. The software system that is provided by the application will manage, among others, Input/output (IO) operations (for data retrieval or storing), authentication, security, and user (human or machine) interaction.



Creating a software application that provides the above-mentioned facilities (eg IO operations, security) to the core computational model, depends on a diverse IT skill set that is usually present only in companies specialised in software engineering. This dependency lengthens the time from development of the computational model to the final software. In fact, the extended time and associated costs of development of a complete computational software service demotivates the companies and hinders the adaptation of software based zero defect manufacturing. Companies, who decide to develop the whole software package themselves, will probably end up with ad hoc solutions resulting in sub optimal products with limited functionality and potential security risks. In ZDMP, the component “prediction and optimisation run-time” (PO run-time) takes care of several issues in the process of transforming models to the final software product. This component consists of a computational core (that contains the computational algorithm) and the service layer which provides the above-mentioned facilities such as REST API and IO services.

What will ZDMP achieve?

ZDMP aims at providing a software ecosystem that facilitates development and deployment of software-based solutions in zero defect manufacturing. The component PO run-time provides computational services to the platform. Multiple instances of the component can run on the platform and other components and applications in the ZDMP platform can delegate their computational tasks to these running instances of the component. For example, a process assurance component can delegate prediction or optimisation operations to several instances of PO run-time that each is specialized, by changing the computational core (algorithm), to perform a specific prediction or optimisation operation. As another example, a digital twin component can request simulation results from an instance of this component that is specialised to perform simulation of any specific production process. In addition to the ZDMP platform, this component can run independently on any desktop computer or OnCloud and provide any client with the result of the computational task that it is specialized for.

This component is modular and taking a new computational model into production, is as easy as copying the new developed package (which performs the calculations) into a designated location. The features of the PO run-time component can be summarised as:

- Access to multiple sources of data for input and output, such as QSL and NoSQL data bases for batch data and Message bus (eg MQTT network) for streaming sensor data
- REST API that provides the user with the ability to interact with the component, set the input source, set output, send algorithm specific parameters and request computation
- Containerised which guaranties hassle-free installation and execution
- Execution of the component can be managed by Kubernetes which allows scaling up or down based on CPU usage
- Benefits from extended facilities of ZDMP platform such as security, user authentication, access to ZDMP platform specific storage and message bus and availability on ZDMP marketplace

In this design pattern the functionality of the core computational module is extended and software reuse is facilitated. This approach, besides accelerating the model design to production life cycle, ensures a more stable and bug free code, which in turn saves a huge amount of time that would be otherwise spent on debugging the code.

ZDMP Links

• **Architecture Component(s)**	Prediction and optimisation run-time
• **Work Package**	WP7 – Process Quality
• **Tasks**	T7.2 – Equipment performance optimisation

References

[1] Amirreza Baghbanpourasl, et al., Virtual Quality control using bidirectional LSTM networks and gradient boosting. Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN) Helsinki-Espoo, Finland

[2] Amirreza Baghbanpourasl, et al., Phase detection of an alternating laser heat source in infrared images using Bayesian recursive filtering. https://doi.org/10.1117/12.2521737