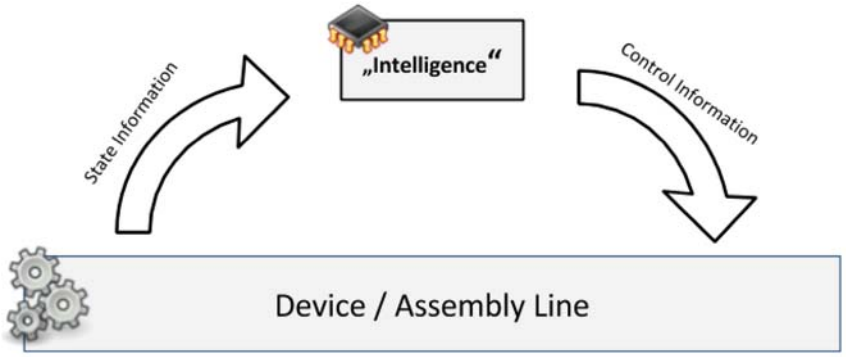
The Importance of Open Industry Standards in the Next Generation Production Systems

Stefan Meissner

Keywords: XJDF, JDF, CIP4, print automation

Industry 4.0, Industrial Internet, Smart Factory, Print 4.0, etc., are topics surrounded by a great deal of discussion in the (graphic arts-) industry. Although there are many interpretations of the exact meaning of these terms, the similarity between most of them is system integration and process automation, controlled by some kind of top level intelligence, making decisions so as to maximize the overall efficiency. For instance, sensors in devices and assembly lines capture the information of the current state. This information is transferred to an intelligent controller or management application. Controllers and management applications make decisions on computing based on the current and historical state information. The result of the computations are decisions which lead to some kind of control information which, in turn, actively influences the behavior of the device or assembly line:



The figure above depicts an abstract view of a modern industrial production system as previously described. In such a system, communication between applications, controllers, and devices is fundamental. Therefore, the question is, how does communication work in detail? Basically, communication is based on interfaces. Next, one has to ask: What is an interface? In general, an interface consists of two parts: the medium for the physical transmission of information, and the data model (data structure) depicting how the information has to be encoded.

Flyeralarm

As the medium is normally given by the environment (e. g. the HTTP protocol) the majority of the work on the specification is dedicated to the data model when developing an interface specification. The data model can be seen as the heart of an interface specification - and an interface specification is a prerequisite for any kind of communication between systems. In order to coerce interaction between systems, these systems have to implement one and the same interface specification. As these interface specifications are so elementary, there are some major conceptual requirements that accompany them:

- Long-Term Stability - Once defined, all changes in an interface specification shall be backwards-compatible. As the goal of an interface specification is to become implemented by many systems, each non-backwards-compatible change will lead to expensive refactoring work on all systems which have already implemented this interface. Due to long-term stability, only extensions shall be allowed in an interface specification.

- Extensibility - As systems and interfaces are usually growing, the data structure needs to be extensible for future requirements. In order to archive this, the data structure has to be designed at a suitable level of abstraction.

  A common issue in the design of data models is that issues and processes initially appear to be simpler than they actually are. Therefore, over time, several definitions for one and the same physical entity or process appear, leading to ambiguities in the interface specification.

- Uniqueness - Uniqueness is another significant conceptual requirement of an interface specification. Having multiple definitions for one and the same physical entity or process leads to unnecessary complexity, incompatibility issues, as well as a required increase in maintenance and development efforts. As explained above, uniqueness very often disappears over time due to interface extensions. A solid data structure shall only allow ONE method to describe one and the same physical entity or process. Either a piece of information can be found at the defined and expected location, or it isn't available at all.

Interfaces are usually designed for long-term purposes. Designing the data model for such an interface requires both broad and detailed functional knowledge, as well as extensive foresight. In the best of all cases, the data model developers will have a crystal ball in order to foresee the new products and the company's development for the next decades - otherwise, the uniqueness of the data model can't be guaranteed while extending the interface over time.

The general problem is that, in order to design a stable data model, broad and detailed functional knowledge is required. Normally, such knowledge isn't available in an individual company, as most companies specialize in just a handful of core topics (narrow and detailed knowledge). Yet another problem is that software development has become agile. This means that there is nolonger a global master plan available for the coming years that defines all the upcoming requirements - I'm glad that.... Modern software development is optimized in short and flexible development cycles (sprints) and focuses on (many) short-term goals rather than on a single (big) long-term goal. The question is, how does one develop a long-term interface specification within an agile development environment?

When designing a data model, people in the industry face the problem of ability vs. stability. The challenge is coming up with a sustainable interface by way of an agile development method which is been successfully put to the test in many companies.

The solution for this kind of problem is open (industry) standards. Open industry standards have been developed for decades and are based on excessive expert discussions between professionals throughout the entire industry.

The result of these long-term discussions is first of all a common and official nomenclature of how to name entities and processes within a specific industry. Based on this common nomenclature, a specific data model is then also accompanied by industry standards. As open industry standards are normally developed by committees whose members are leading companies, vendors and consultants, the data model is developed based on a very broad and detailed view of the industry.

From the perspective of the committee, this broad viewpoint allows for the design of a data model at the right level of abstraction which is able to satisfy the major conceptual requirements mentioned above: Long-Term Stability, Extensibility and Uniqueness.

As many open industry standards have already been in existence for some time, the stability of many parts of the data model have also been tested and confirmed by real-world implementations.

When working with open industry standards, a good strategy is to keep the data structure, remove everything not yet needed, and then reintroduce extracted parts on a step by step basis as needed. Using this method, stable long-term interfaces can be built in an agile way. Open industry standards can be seen as the bridge between agile software development and the long-term stability of interfaces.

In the graphic arts industry, XJDF (Exchange Job Definition Format) defines such an open industry standard. "XJDF is a technology that allows systems from many different vendors to interoperate in automated workflows. While it is technically an XML software specification, it is more importantly a means of connecting multiple vendor solutions with a workflow solution for automation. XJDF is the first major version update of JDF."[1] JDF, the Job Definition Format, is the successor of PPF, the Print Production Format, and was published in 2001. PPF itself was published in the mid nineties. As one specification has been built upon the other, the data model accompanying the XJDF Specification is built based on more than 20 years of expert discussions and implementation experience. As a result, the XJDF data model has become very stable in many parts, and, for this reason, it is the perfect prerequisite for interface specifications within the graphic arts industry.

XJDF is maintained by the CIP4 Committee. The CIP4 Committee consists of a broad range of members all over the graphic arts industries such as printers, vendors for prepress, press and postpress, consultants, universities, etc.

In short: Independent of how the terms Industry 4.0, Industrial Internet, Smart Factory, and Print 4.0 are interpreted and implemented, XJDF provides the perfect data model for the next generation of print. There is no other standard in the graphic arts industry that is as powerful and/or as comprehensive as XJDF is.

References

1. XJDF Specification 2.0 - IP Review 3 - "Introduction"