Review paper http://doi.org/10.24867/JGED-2018-1-005

Smart Factory: JDF and XJDF

ABSTRACT

In this paper, the relation between "Industry 4.0" and the metadata formats JDF/XJDF is discussed. While "Industry 4.0" is a term for a certain method of production in general, the XML-based data formats JDF/XJDF are commonly considered being technologies for implementing Industry 4.0 production in the graphic arts industry. The paper shows that the original architecture behind JDF, however, is more compatible with Industry 4.0 that the new XJDF format. For this purpose, some important features of these two data formats are outlined.

Thomas Hoffmann-Walbeck

Stuttgart Media University,
Stuttgart, Germany

Corresponding author:
Thomas Hoffmann-Walbeck
e-mail: hoffmann@hdm-stuttgart.de

KEY WORDS
JDF, XJDF, PDF, Metadata, Print Production, Industry 4.0

First recieved: 31.08.2017.
Accepted: 31.10.2017.

Introduction

Smart Factory is a general research concept concerning automation of industrial production processes. This term derives from the "Industry 4.0" initiative, the future project of the German government. One of the main features is the data exchange between production machines based on internet technologies.

For almost two decades, the JDF/JMF format dominated these interfaces in the graphic arts industry. Therefore, in section 3 the basics of data structure of the Job Definition Format (JDF) is presented. It will be shown that JDF constitute important parts of industry 4.0.

However, it is very likely that JDF will be replaced by a new format in the future, the "XJDF". Section 4 will outline the differences between the two formats, the motivations for the redesign of JDF and their relations to "Industry 4.0".

Smart Factory and Industry 4.0

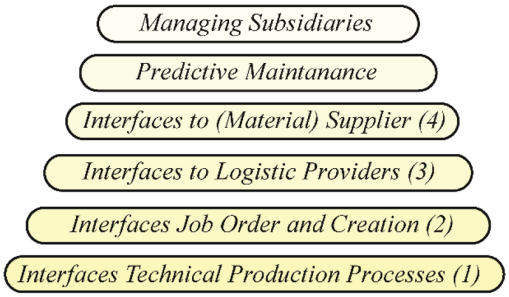
One of the biggest challenges in the current industrial production is flexibility. In the past, the productivity of mass production has been risen by fixed automation solutions. Nowadays, however, the production of many variants of customized products in small series becomes increasingly important. A paradigm shift is expected- from centralized control toward a flexible, decentralized coordination of autonomous operations. The term "Industry 4.0" defines the vision of such production environments, where customer orders control their individual production, book their production machines and their material and finally organize their delivery to the customer (Spath et al., 2013). In such manufacturing plants ("smart factories") the machines and tools might interconnect, organize and configure themselves independently in future times. See also (Federal Ministry of Economic Affairs and Energy, 2017).

Interrelating the general concept of "Industry 4.0" to the graphic arts industry, the following four important classes of interfaces are involved according to the statement above.

1. Technical production processes on the shop-floor of a Print Service Provider (PSP),
2. Print Buyer (PB) and PSP,
3. PSP and supplier of material and services,
4. PSP and logistic provider.

There are, in fact, some more areas, that are also considered being part of Print 4.0. In practice, they depend on each other, which is graphically illustrated in figure 1. The pyramid shows a certain hierarchy. In this paper, only the vital interfaces in the production processes at the PSP are discussed. The other areas of automation are outlined e.g. in (Hoffmann-Walbeck & Riegel, 2017).

This metamorphosis of production methods might be cutting-edge technology for the industry in general, for the graphic arts industry, however, it is not. For most PSPs diversity of variants and short run-lengths are daily routines. Different technologies of metadata like XMP (ISO, 2012), Web services, CSV are used for this matter, but JDF/JMF still is the most elaborate and dominant communication model.



» **Figure 1:** *Sub-topics of Industry 4.0 in Graphic Arts Industry*

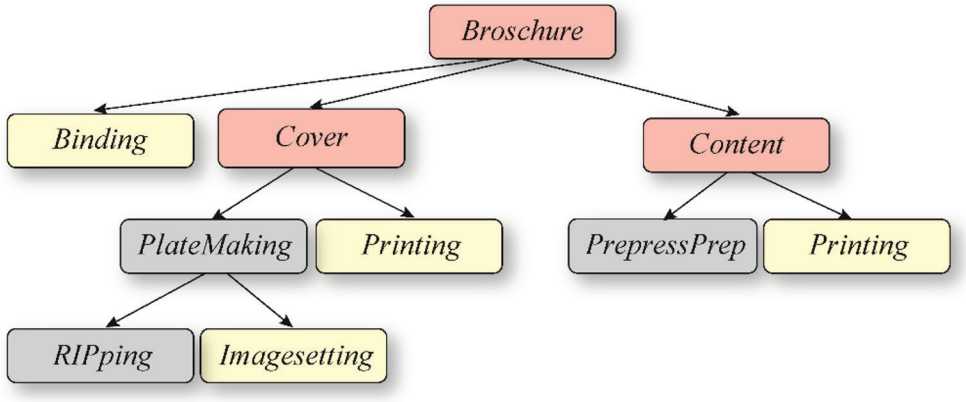
Job Definition Format (JDF)

The first specification of the Job Definition Format (JDF) (CIP4, 2013) had been published in 2001 and complies with the concept of the smart factory in the printing industry ("Printing 4.0"). To show that, let us recap some of the important structures of the format (See also (Hoffmann-Walbeck & Riegel, 2012).

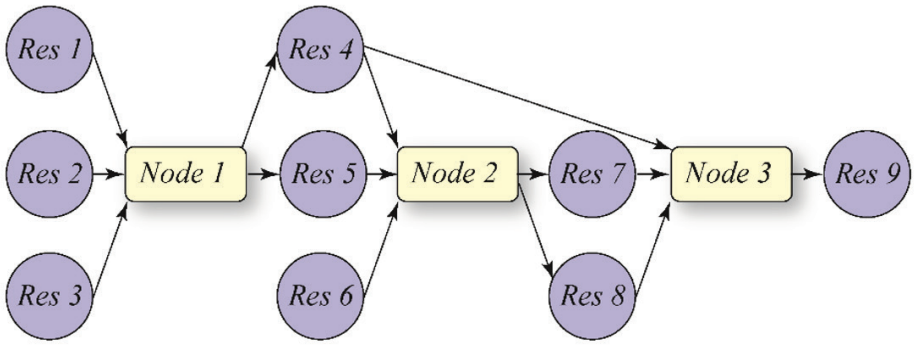
With JDF, one can describe the intended product itself, as well as the processes that are needed to produce the product and its product parts. Processes can be encapsulated into "Process Groups". All product (parts), process groups and processes are described by XML nodes with the tag-name "JDF", which are actually called "JDF nodes". The JDF nodes are structured as a tree, whereas the root typically represents the product itself and the children the product parts and/or processes and process group nodes. The process nodes and/or process group nodes that are descendants of a product node describe the processes concerning the production of the product or product part. Figure 2 shows a fictive and simple example. Here, the Product Nodes are highlighted in red, the Process Groups in gray and Process Nodes in yellow. The tree does not reflect any process order.

Typically, a Management Information System (MIS)- i.e. a software for managing customer orders- initiates writing JDF data and generates the product node and product part nodes for a job order. It can also define processes and process groups, but since the MIS would not know much about the technical details of the required production processes, it can do that only in a superficial manner. During the production, other devices will add the necessary details as well as the results of the process execution (status, used resources and the like). That is, the JDF data enlarges during production. In the end, the JDF will contain all settings of the production workflow, provided by all components that are JDF savvy. Normally, such a JDF file contains dozens of JDF nodes.

The JDF workflow representation is based on the "process-resource-model" (PRM). A process is an activity, while resources are either physical entities (paper, plates, ink…) or electronic data (PDF files, images, profiles, parameter sets…). These resources are either input or output resources for a JDF node. An input resource of a process node is an entity that is needed for the execution of the process; an output resource is the outcome of the process. An input resource for a product node is called a "Intent Resource", because it describes the customer's intention of the printed product. Figure 3 represents a small segment of a fictive PRM.



» **Figure 2:** *JDF nodes*



» **Figure 3:** *Process-Resource-Model*

The PRM is merged into the node tree. Each node contains a reference to its input and output resources. Many resources are referenced by two or more JDF nodes. An output resource of a process node, for example, may be an input resource for another process (like Resource 4, Resource 5 and Resource 7 in figure 3). The main idea of automation using JDF metadata is that devices can execute processes automatically, if all input resources are available, the device is idle and the production window (time span) is met. However, since it is not advisable to have different copies of a resource in the JDF data, a resource can be "almost anywhere" in the JDF data and is not necessarily placed inside a JDF node that refers to the resource. Inside a JDF node, however, the references of this node must be specified via an unambiguous identifier. This leads to a highly structured net of nodes, resources and references between the two.

One way to communicate JDF data is that it can be sent (as a file) from device to device. Each device then extracts the information from the file (typically one or more JDF nodes and all resources that are referenced) and adds new entries into the file (e.g. operational data). This linear fashion of passing JDF data around can be enhanced by parallel distribution, since JDF allows locking parts of the data, so that two different devices need not extend JDF data concerning the same data part.

This architecture is getting close to the ideas of industry 4.0, which has been postulated more than a decade after the specification of JDF. Even plug-and-play features are part of the JDF specification (though still far away from implementation) as well as descriptions of device capabilities.

Sending entire JDF files from device to device, however, is not common (any more). Most devices are, in fact, controlled by one or several JDF controllers. These controllers pass along individual and appropriate JDF data to each device they control and integrate the updated data which they receive from the devices. This actually relieves the devices from the burden of understanding and managing the entire workflow and the controller might have other means (like private databases) to determine the production workflow. This architecture, however, moves away from the vision of Industry 4.0. All devices, however, still must be prepared to receive a full JDF workflow description and must be able to extract all relevant information out of it by its own means.

XJDF

XJDF is a redesign of JDF that is currently specified (CIP4, 2017). See also (Meissner, 2017). JDF is a very powerful, but in the same time quite a complex language as explained. There are node hierarchies to be observed, there are resources to be searched for in the data and there is a merger of product and process descriptions. Furthermore, JDF even is not even pure XML, e.g. properties of so-called "Partitioned Resources" can be inherited. Thus JDF is hard to interpret, expensive to implement and prone to incompatibilities. Because of this complexity, change orders and ganging jobs are not so easy to implement in JDF.

In particular, since a controller usually defines the production workflow internally in private databases anyway, there is no need to describe the entire production workflow in XML as well. Also, JDF based production environments tend to become slow, when a huge amount of jobs are processed simultaneously, since reading and analyzing XML text files are less efficient than, for example, querying a database. With XJDF, an overall workflow description has been abandoned altogether. XJDF will become mainly an interface language between a controller and a device. In order to state this fact more generally: XJDF is a communication protocol between two applications. This makes things a lot easier, especially for the devices. Any device receives typically only a single process node now, which contains all resources it needs for executing the process. References to resources outside the process node are not required any longer. Change orders should be easier to handle, since now it is only necessary to send new versions of XJDF data to those devices that needs to know about it. The XJDF structure is very flat (compared to the JDF structure). A XJDF node for a specific process typically contains a product list and resource sets (See figure 4).

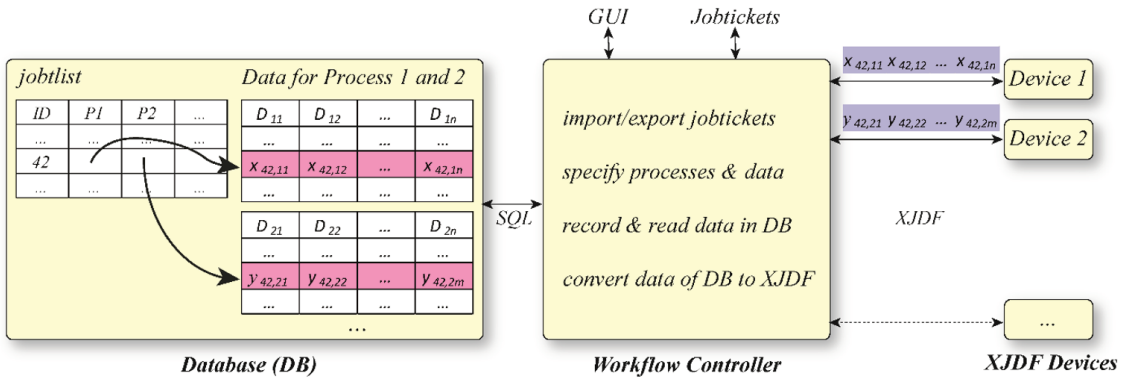
```
<xjdf:XJDF JobID = "JOB-4711" Types = "Folding" xmlns:xjdf = "http://www.CIP4.org/JDFSchema_2_0">

      <xjdf:ProductList>
            <xjdf:Product Amount = "100" DescriptiveName = "Leaflet-1">
                  <xjdf:Intent Name = "ColorIntent">
                        …
                  </xjdf:Intent>
                  …
            </xjdf:Product>
      </xjdf:ProductList>

      <xjdf:ResourceSet Name = "Component" Usage = "Input">
      …
      </xjdf:ResourceSet>
      <xjdf:ResourceSet Name = "Component" Usage = "Output">
      …
      </xjdf:ResourceSet>
      <xjdf:ResourceSet Name = "FoldingParams" Usage = "Input">
      …
      </xjdf:ResourceSet>
      <xjdf:ResourceSet Name = "Media">
      …
      </xjdf:ResourceSet>
      …
</xjdf:XJDF>
```

» **Figure 4:** *Snippet of XJDF Sample Code of process "Folding"*

The product list contains one or several products or product parts from the PB's point of view. The properties of the intent products are defined by intent-elements. The resource sets describe resources that are required by the process. In addition, XJDF is pure XML, so that standard XML tools can be used for handling.

Figure 5 shows a simplified architecture of an XJDF configuration. The Workflow Controller keeps a job list in a database and for each job a set of parameters are stored for each process (device). In figure 5 only two processes are listed. The controller specifies which processes are needed for a given job as well as the data that are required for each process. This information typically will stem from some job ticket that the controller imported, by default values or by the entries that an operator has filled in a graphical user interface. The manufacturer of the controller is free to design the database (or any other means of storage) according to his own wishes. The data in the database could be specified like the resource elements in the XJDF, but they do not have to be compatible with XJDF at all. The controller only has to convert its own data into a XJDF resource list for the XJDF device. To illustrate this fact, we modified the color of the data in the DB and in XJDF.

With XJDF, there is no more fixed connection between a single product description and the definition of production processes like it was with JDF. The above-mentioned product list in XJDF can contain descriptions of several independent products. This will make the implementation of ganging jobs easier. Furthermore, it reflects the situation of Web-to-Print configurations much better. Moreover, in future times, one might store product description outside of XJDF altogether. The product description might be stored within a Page Description language. In (CIP4, 2015) a set of PDL-independent standard metadata keywords are defined, which enables product description. In PDF, this metadata can be stored by the DPART-element of PDF/VT (ISO, 2010) respectively PDF 2.0 (ISO, 2017). XJDF might then only contain production resources.



» **Figure 5:** *Configuration of Controller and Devices applying XJDF*

Conclusions

The JDF specification has anticipated the vision of "Industry 4.0" in parts. The current implementations of JDF-based workflow software, which predominantly can be found in offset and in digital printing, moved away from the original idea of decentralization. This will be even more so for XJDF. Since XJDF, however, is much easier to implement in a production workflow that JDF, the format XJDF might become an important industry standard or might even replace JDF altogether in the future nevertheless. It will be interesting to observe, if the vision of "Industry 4.0" will also recollect the concept of centralization in the years to come.

The shift of production description from JDF/XJDF towards PDF, however, supports the interface between PB and PSP according to the mission statements of Printing 4.0. This concept, however, has not yet been implemented so far.

The other data formats XMP and CSV mentioned in this paper are container formats, which usually only transport private data between different software modules of a single manufacturer. There are no standard descriptions for production processes or resources concerning those formats.

References

International Organization for Standardization (2010) ISO 16612-2:2010. *Graphic technology - Variable data exchange - Part 2: Using PDF/X-4 and PDF/X-5 (PDF/VT-1 and PDF/VT-2).* Geneva, ISO.

International Organization for Standardization (2012) ISO 16684-1:2012. *Graphic technology - Extensible metadata platform (XMP) specification – Part1: Data model, serialization and core properties.* Geneva, ISO.

International Organization for Standardization (2017) ISO 32000-2:2017. *Document Management - Portable Document Format - Part 2: PDF 2.0.* Geneva, ISO.

Federal Ministry of Economic Affairs and Energy, Federal Ministry of Education and Research. (2017) *Plattform Industrie 4.0.* Available from: http://www.plattform-i40.de/I40/Navigation/EN/ [Accessed 1st May 2017].

Hoffmann-Walbeck, T. & Riegel, S. (2017) Advances in Digital Production Workflows. In: Media Technology Department of the Technology Faculty at Kauno kolegija/University of Applied Sciences. *International Scientific-Practical Conference Innovations in Publishing, Printing and Multimedia Technologies 2017.* Kaunas, Kauno Kolegija/University of Applied Sciences. pp. 30-34. Available from: http://conference.media.kauko.lt/files/2014/01/Konferencija_2017.pdf

Hoffmann-Walbeck T. & Riegel, S. (2012) JDF Workflow: *A Guide to Automation in the Graphic Communications Industry.* Warrendale, Printing Industries of America.

Meissner, S. (2017) *XJDF – Exchange Job Definition Format.* Regensburg, Ricebean.net.

Spath, D. [ed.], Ganschar, O., Gerlach, S., Hämmerle, M., Krause, T. & Schlund, S. (2013) Produktionsarbeit der Zukunft – Industrie 4.0. *Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.* Stuttgart, Fraunhofer Verlag. Available from: http://www.produktionsarbeit.de/content/dam/produktionsarbeit/de/documents/Fraunhofer-IAO-Studie_Produktionsarbeit_der_Zukunft-Industrie_4_0.pdf [Accessed 1st May 2017].

The International Cooperation for the Integration of Processes in Prepress, Press, and Postpress Organization-CIP4 (2013) *JDF-Specification, release 1.5.* Available from: https://confluence.cip4.org/display/PUB/JDF [Accessed 1st May 2017].

The International Cooperation for the Integration of Processes in Prepress, Press, and Postpress Organization CIP4 (2017) *XJDF Specification, 2.0-RC-1-43.* Available from: https://confluence.cip4.org/display/PUB/XJDF [Accessed 29th October 2017].

The International Cooperation for the Integration of Processes in Prepress, Press, and Postpress Organization-CIP4 (2015) *ICS — Common Metadata for Document Production Workflows.* Available from: https://confluence.cip4.org/display/PUB/Common+Metadata+for+Document+Production+Workflow+ICS [Accessed 29th October 2017].