CIP4 Technologies I
Job Definition Format

Distribution of JDF
JDF Nodes
Gray Boxes,
JDF Resources, ResourceLinks
Gray Boxes
Partitioned Resources

Thomas Hoffmann-Walbeck
Metadata Communication Hub

PP's Partners
- Customer
- Paper Supplier
- Plate Supplier
- Logistic Services
- Packing & Shipping
- Die Cutting

PrintTalk
- JDF
- XMP
- JMF
- CFF2
- DB
- private XML
- Exif
- CSV
- XJDF
- XJMF
- PPF
- PJTF

Job Scheduling
- Reporting
- Preflight
- Plate Making
- Printing
- Cutting
- Stitching
- Collecting
- Folding

PP Management
- PP's Partners
- PP Processes
Central Statements about JDF

- In each JDF job ticket there are “normally” product and production descriptions for a single job.
- JDF is based on the process-resource model.
- The main accomplishment are the specifications of most processes and resources for the graphical industry.
- Several processes can be subsumed to a ProcessGroup.
- JDF resources are stored in a ResourcePool; A ResourceLink is stored in ResourceLinkPool.
- Gray Boxed are ProcessGroups that are not complete.
- Resources can be partitioned.
Information Exchange with JDF/XJDF

MIS

Page Size
Sheet Size
Colors
Printing substrate
Run length
Order information
(Imposition Scheme)

PostPress

Ink zones presetting
Previews
Cutting positions
Folding positions
Operational data

PrePress

Press

Page Size
Sheet Size
Colors
Printing substrate
Run length
Order information
(Imposition Scheme)

Operational data
Sequential Architecture

- A single JDF file per job passes from one JDF module to the next e.g., via hot folder.
  - No database, original concept, somehow outdated.
  - JDF “gains weight” during production.
- In reality, processes are not sequential.
  - Overlaps (pipes), parallelism (data consistency).
- Quite complex if level of JDF-integration is high
  - Difficult to maintain de-central data storage.
Central JDF Storage For All Modules

- Some software is in charge for “rw” of JDF, e.g., MIS.
- Processes may not use/update JDF data at the same time.
  - Spawning & merging: A JDF part rw-spawned only once at a time
  - Nevertheless: Each module must be prepared to extract JDF nodes it needs from entire job ticket.
(X)JDF is just a communication protocol.

(X)JDF, (X)JMF
Private DB access
Hierarchy of JDF Nodes

- Typically, MIS initializes a JDF record for each print job.

- A JDF-Job is defined by nodes:
  - Product nodes,
  - Process group nodes,
  - Process nodes.

- JDF nodes are described by XML-elements and XML-attributes.
  - A node is an XML element with the tag “JDF”: <JDF…>.
  - “Arrows” between those elements represent JDF sub-nodes.
  - The node type is stored in an attribute, e.g., Type = “Product”.

Thomas Hoffmann-Walbeck
Example Tree Structure from Production
Attributes of a Node

- Attributes vary for different node types.
- Each JDF node has a unique ID. This is used to refer to the node.
- For processes the specific kind of process is specified, e.g., `Type=“Scanning”` or `Type=“Trimming”`.
- The attribute `Status` denotes the status of a JDF node.
  - E.g. `Waiting`, `Ready`, `InProgress`, `Stopped`, ...
- Besides attributes, a JDF node has often many sub-elements.
  - More JDF nodes or other XML elements according to the spec.
(Almost) each node has input and output resources

- Some of those are mandatory, most are optional (marked by “?”).
- The interpretation of resources for product nodes differs from that for processes and process groups.
Resources for Product Nodes

- **Output** resources of a *Product* node is a *Component* (= representation of the final product or a product part).

- **Input** resources are either Components (= product parts) or *Intent* resources.

- **Intent** resources are input resources. They describe the intents of the print buyer, e.g., binding types, number of pages, finishing options, colours, paper, delivery, …

  »A range of acceptable values for various features of the product can be defined, e.g., *Weight* = “150“ or *Range* = “130 ~ 180“ in *MediaIntent*. 
Resources for Processes

Resources provide product or process details.

» Sometimes hard to specify.

```xml
<BinderySignature>
  Class="Parameter"
  ID="_4711"
  BinderySignatureType="Fold"
  BindingEdge="Left"
  FoldCatalog="F4-1"
  NumberUp="2 1"
  Status="Available"
  ...
</BinderySignature>
```
How are Resources Saved?

- Process nodes have input and output resources.
  »Same holds for Product nodes and ProcessGroup nodes.

- The input and output resources of a node are not necessarily stored inside the node.

What could be the reason for that?

Hint: Remember transitional resources!

- Answer: Otherwise, multiple copies of resources would be needed.

- Thus, there must be some other mechanism to define which resources are needed for a node!
Nodes and Resources

Product
- R₁
- R₂
- R₃
- R₄
- R₅
- R₆

Product Part
- R₇
- R₈
- R₉
- R₁₀
- R₁₁

Process
- R₁₆
- R₁₇
- R₁₈

Process Group
- R₁₉
- R₂₀
- R₂₁
- R₂₂

Process
- R₂₈
- R₂₉

Process
- R₃₀

Product Part
- R₁₂
- R₁₃
- R₁₄
- R₁₅

Process
- R₂₃
- R₂₅

Process
- R₂₆
- R₂₇

[Diagram showing relationships between nodes and resources with input and output arrows]
ResourcePool and ResourceLinkPool

- Resources are accumulated in ResourcePools.
- Each ResourcePool is a sub-element of a JDF node.
- Not every JDF node needs to have a ResourcePool.
- For a JDF node, the reference of an input/output resource is a ResourceLink (defined via a resource ID).
- ResourceLinks are collected in a ResourceLinkPool.
- Each ResourceLinkPool is a sub-element of a JDF-Node. It contains all ResourceLinks for that JDF node.
- A resource of a JDF node can be located in a ResourcePool of a different JDF node.
Structure of a JDF file

```
<?xml ...?>
<jdf ...
...
<ResourcePool>
...
</ResourcePool>
<ResourceLinkPool>
...
</ResourceLinkPool>
<jdf>
...
</jdf>
```

- **Resources**
- **Links of the JDF node to its resources**
- **Further JDF node(s)**

JDF root element
Each node can contain information about execution, responsibility for the node and consequences for a delay.

» FirstStart, LastStart, FirstEnd, LastEnd,
» DueLevel (values: Trivial, Penalty, JobCancelled),
» JobPriority,
» Route (URL of controller, which is supposed to process node),
» Employee.

```
<JDF ID="ScreenTest" Type="Product" JobID="ScreenJob" Status="Waiting"...>
  <ResourcePool>
    <NodeInfo LastEnd="2012-12-24T06:02:42+01:00"/>
  </ResourcePool>
</JDF>
```
Execution of a Process Node

- Device / Controller executes a process, if
  - Device/controller can handle the node type and is available,
  - Status of the node is ready or waiting,
  - All input-resources have status available,
  - The production time window in NodeInfo fits.

![Process Node diagram](image)
A Gray Box is a special ProcessGroup node for which not all details are specified yet.

» "A Gray Box specifies a loose combination of several processes with a specific goal, e.g., “PlateMaking”.

» A Gray Box does not specify all processes or all resources - except for output resources.

» Usually, a MIS generates Gray Boxes.

During the production the production details are added, so that at the end the Gray Box can be dissolved and normal JDF process nodes (or a normal ProcessGroup or a Combined Process).
Gray Boxes Life-Cycle

- Gray Boxes cannot be executed!

- During the production new processes (JDF nodes) will be generated with the process names.
Why *GrayBoxes*?

- A MIS normally only has data that is needed for calculating the price such as
  - **Administrative data**: customer details, JobID,
  - **Job data**: Number of pages, page size, colour, type of binding,
  - **Production data**: Imposition schemes, paper, printing press.

- A MIS-System normally has no information about:
  - Name of the content-file(s),
  - Process details about trapping, screening, marks, …

- The missing data will be added either by default-values of the workflow modules, by an operator manually, internal data base or as a result of an earlier process.
Partitionable Resources

- It is common to see things in their entirety or in subsets.
  - e.g., means of transportation->car->car brand->"My car".

- What is easy to describe colloquially must be defined precisely in a formal language.

- Structuring resources that consists of several physical or logical units.
  - E.g., printed sheets or printing plates.
  - E.g., all plates for a job versus a plate for a specific separation, for front or back of a specific sheet.
  - Without partitioned resources each sheet/plate need be a resource and we would end up with thousands of resources.
Partitioned Resources

- Partitioned Resources are often separated by the value of the attribute `PartIDKeys`.

  » In this example: `ExposedMedia` are printing formes

  `PartIDKeys="SignatureName SheetName Side Separation"`