

JDF Technical Overview

Job Definition Format

www.job-definition-format.org

Dr. Rainer Prosi

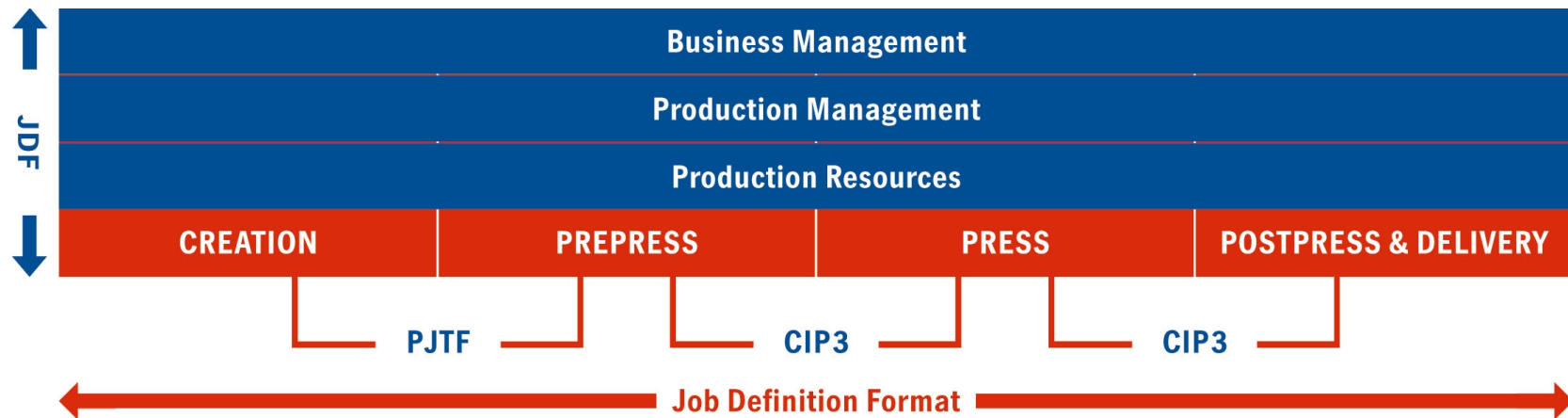
Heidelberger Druckmaschinen AG

Table of Contents

- JDF Overview
- JMF Messaging
- Encoding
- Comparison with PJTF
- Comparison with CIP3 PPF

Scope Of JDF

- Horizontal Job Description
 - Job Ticket
- Vertical Communication
 - Messaging



JDF Properties

- **JDF is a Data Interchange Format Specification**
- Encoded in XML
- Extensible
- Process Modeling with **Nodes** and **Resources**
- Adobe PJTF: simple 1--1 mapping to JDF Resources
- CIP3 PPF: simple mapping to JDF Resources
- JDF Job Definition + JMF Messaging define the JDF Framework

Content vs. Meta-Data

- JDF is independent of the Content Description Language
 - PDF
 - PS
 - PPML (**P**ersonalized **P**rint **M**arkup **L**anguage)
 - ...
- JDF contains no Content Data
 - Content data is referenced with URL links
- JDF contains Production Data

JDF High Level Elements (I)

- JDF Node
 - Specifies a Product, Process or group of Processes
 - Modifies/Consumes/Creates Resources
 - May contain further nested JDF Nodes
- Resource
 - Parameters or Logical Entities
 - Physical Entity
 - Component
 - Handling Resource
 - Consumable
 - Implementation- Device or Employee
- Resource Links
 - Bind a Resource to a Node

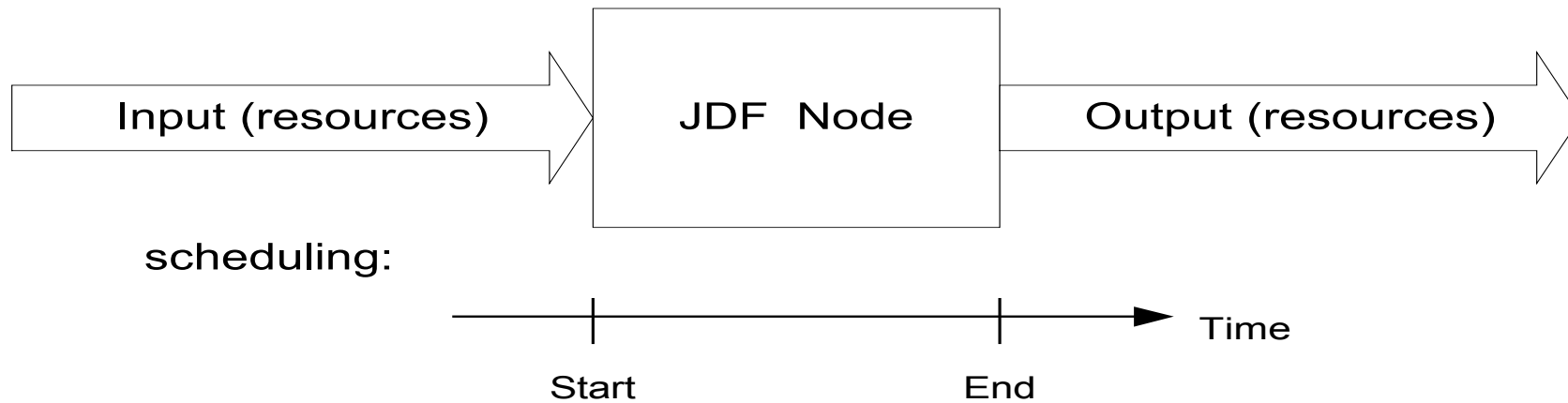
JDF High Level Elements (II)

- NodeInfo
 - Logs scheduled, planned job properties
- Audit Object
 - Logs actual, post-facto job properties
- Customer Object
 - Customer + Delivery Address
- Comment
 - Internationalization
 - Path / Box annotation
- Localized to local node and all child nodes
 - Subcontracting

The JDF Node (I)

- One JDF node type for Products and Processes
 - Allows Branching and Merging of JDF for subcontracting, parallelizing
 - The Question “Is it a sheet or a sheet-making process?” is irrelevant.
- Less Precise Product Intent at the Root
- Workflow Groups in between
- Detailed Processes in the Leaves
- Job + Job Part Identification

JDF Node and Resource interaction



scheduling:

JDF Node -- Simple Example

```
<JDF ID="HDM20000824112251" Type="Product" JobID="HDM20000824112251"
  Status="waiting" Version="0.9">
  <NodeInfo/>
  <CustomerInfo/>
  <ResourcePool>
    <SomeInputResource ID="Link0002" Class="Parameter" Status="available"/>
    <Component ID="Link0003" Class="Quantity" Status="unavailable"
  DescriptiveName="SomeOutputResource"/>
  </ResourcePool>
  <ResourceLinkPool>
    <SomeInputResourceLink rRef="Link0002" Usage="input"/>
    <ComponentLink rRef="Link0003" Usage="output"/>
  </ResourceLinkPool>
  <AuditPool/>
</JDF>
```

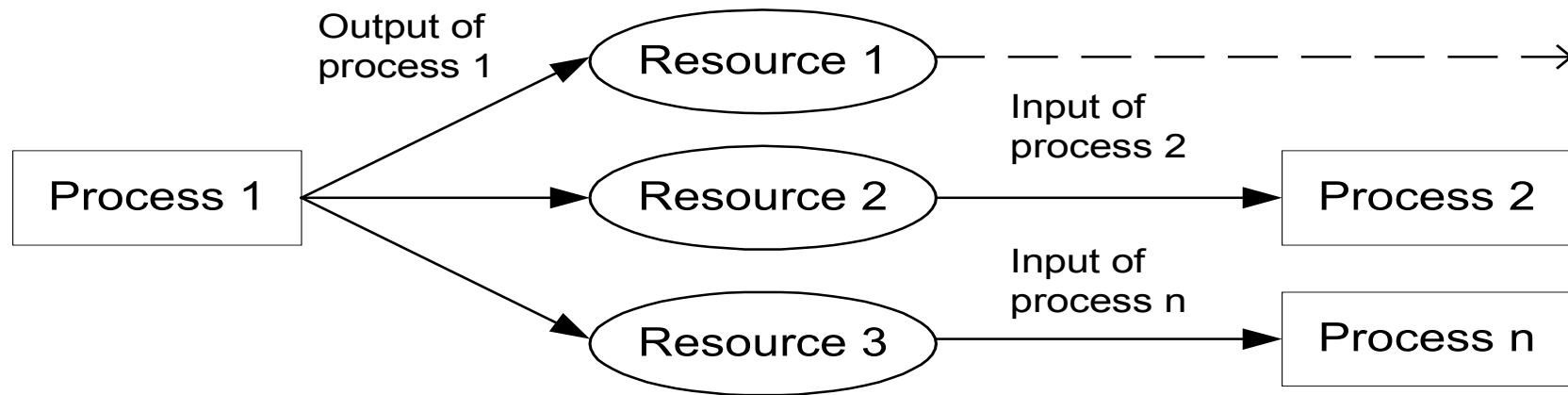
Job Description Models

- Product Definition
 - No Process
 - Abstract
 - Segmentation by Product Components
- Serial Processing
- Parallel Processing
- Overlapping Processing
 - Pipes
- Iterative Processing
 - Informal Iterative Processing using Draft Resources
 - Formal Iterative Processing additionally using JMF Messages

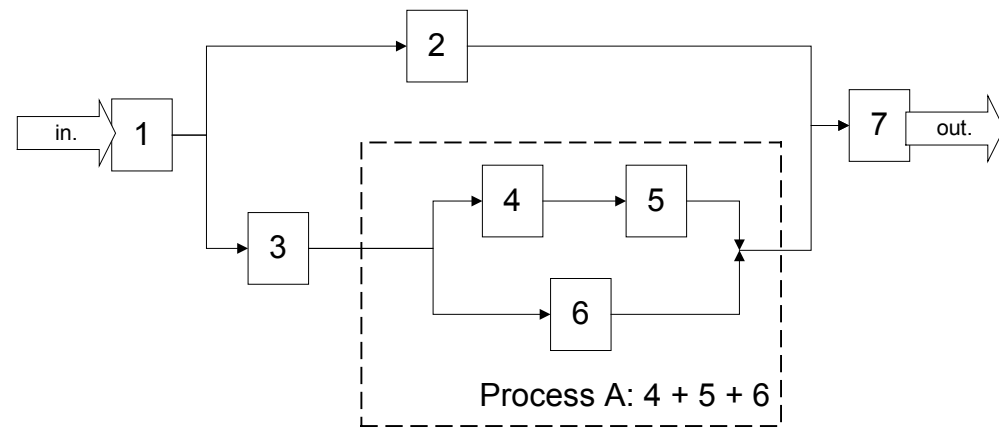
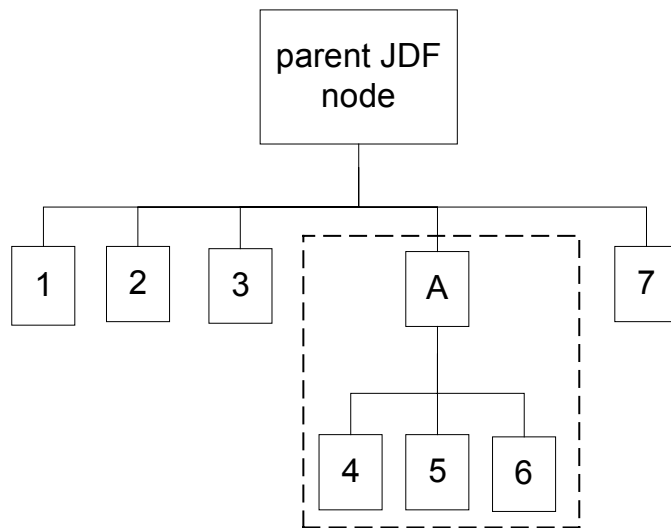
The JDF Node -- Process Model

- A Node is executable when all required input resources are available
- Define arbitrary workflow sequencing
- Link one resource to multiple nodes.
- Node dependencies allow Process configuration
 - A proof node can create an ApprovalSuccess Resource which is a required input resource for a printing node.

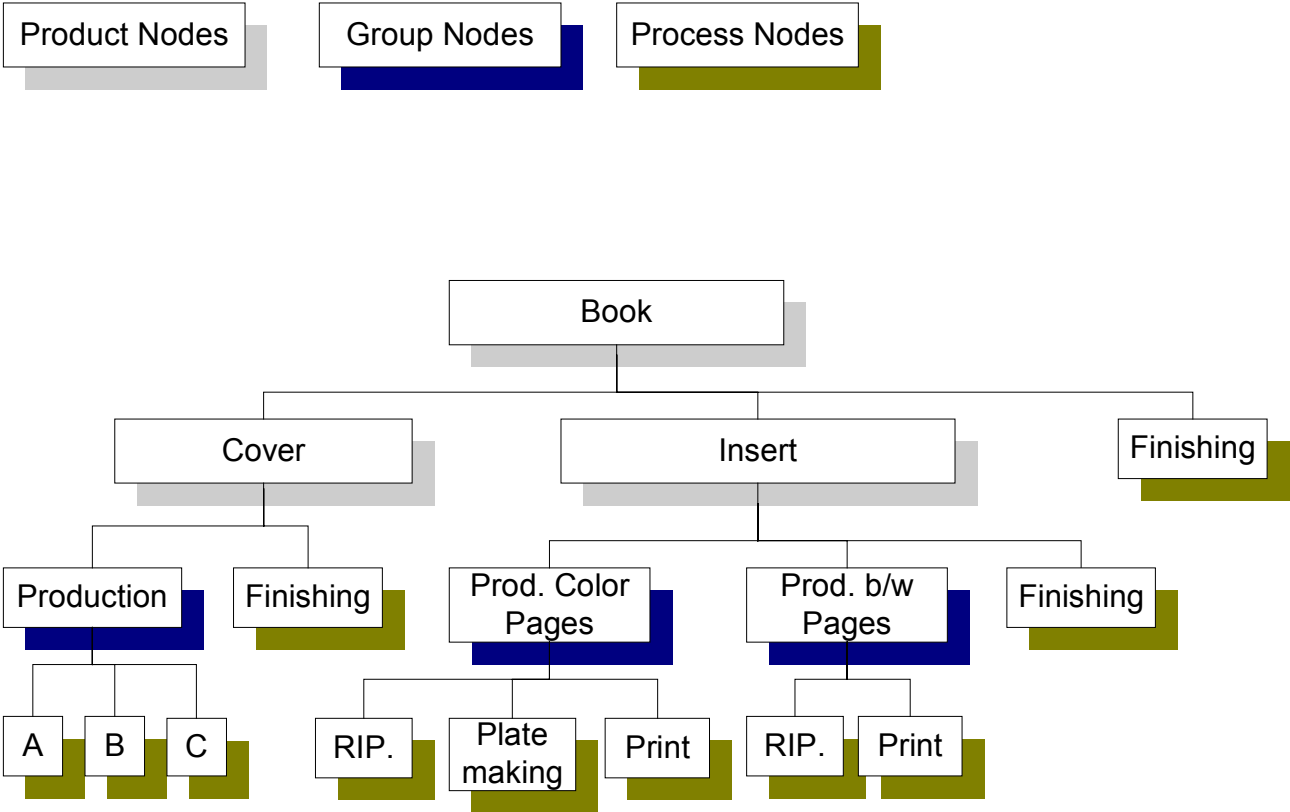
JDFs Linked via Resource Links



JDF Tree / Network Structure



JDF Job Example



Combination of JDF Nodes

- Define a limited number of “atomic” Processes.
- Combine multiple predefined processes into one process, e.g.:
 - inline finishing = printing + folding +cutting;
 - in-RIP trapping = trapping + RIPping
- Two types of Combination Nodes
 - **Combined Node**: All internal interfaces are hidden
 - Smart multi-function device
 - **ProcessGroup**: Internal nodes are accessible
 - Workflow group in a department
 - Subcontract

JDF Resources (I)

- Based of PJTF and CIP3 PPF
 - Sheet definition, Colors : PJTF
 - Press, Finishing: CIP3
 - Extensions where appropriate.
- Internal to JDF
 - Parameters
 - URL definitions
- External Links to well-defined Formats
 - Thumbnails,
 - ICC Profiles
 - Content Data

Resource Classes

- Parameter
 - Process Parameters
 - Content File reference
- Consumable
 - paper; ink; raw plate
- Handling (Reusable)
 - developed film
- Quantity
 - Printed or processed Material
- Implementation;
 - Devices (limited to string + family in v1.0)
 - Operators: id, shift, role
- Selector
- Placeholder

JDF RunList Resource Simple Example

```
<RunList ID="Link0003" Class="Parameter" Status="available">
  <Run Pages="0~10">
    <LayoutElement ElementType="document">
      <FileSpec FileName="panrt17a.pdf" MimeType="app/PDF"/>
    </LayoutElement>
  </Run>
  <Run Pages="12~-1">
    <LayoutElement ElementType="document">
      <FileSpec FileName="MyFile.pdf" MimeType="app/PDF"/>
    </LayoutElement>
  </Run>
</RunList>
```

Partitioned Resources

- Inheritance of common Data
 - Overwrite defaults
- Predefined Partition Keys:
 - Sheets
 - Separations
 - Amounts
 - Tiles
 - Versions
- Access individual parts of a large resource
 - Only the yellow plate of the front surface of sheet 17
- Mechanisms for Parallel Processing of Partitioned Resources

Partitioned Ink Resource Example

```
<Ink ID="Link0007" Brand="ProcessBrand" Class="Consumable"
  Status="available" MediaType="Coated" PartIDKeys="Separation">
  <Ink Separation="Cyan">
    <Colorant CMYK="1 0 0 0"/>
  </Ink>
  <Ink Separation="Black">
    <Colorant CMYK="0 0 0 1"/>
  </Ink>
  <Ink Brand="SpotBrand" Separation="Heidelberg Spot Blue"
  Status="unavailable" >
    <Colorant CMYK="0.7 0.3 0.3 0.7" ColorantUsage="spot"/>
  </Ink>
</Ink>
```



ResourceLink

- Bind a Resource to a Node
 - A Resource is NOT bound to the JDF that contains it unless it is linked by a ResourceLink.
- Define Resource Usage (input or output)
- May Link to a Subset / Part of a Resource
- May Contain Pipe control meta-data.
- Live in ResourceLinkPool of a JDF node.
- Name is derived from the linked resource
- Allow reuse of Resources by multiple processes
 - One resource may be linked by multiple ResourceLinks

ResourceLink Examples

- Simple

```
<ScanParameterLink ID="ScanLinkID" />
```

- Quantity

```
<MediaLink ID="MediaLinkID" Amount="6642" />
```

- Part (Links to previous example)

```
<InkLink ID="InkLinkID" >
```

```
<Part Separation="Heidelberg Spot Blue" />
```

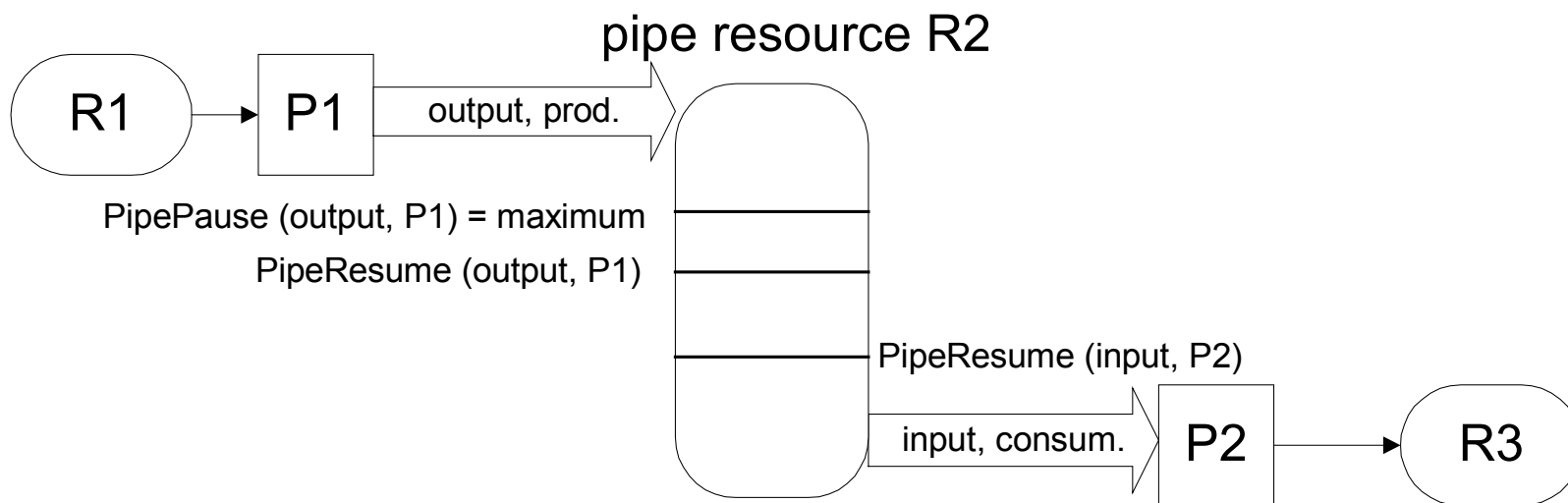
```
</InkLink>
```



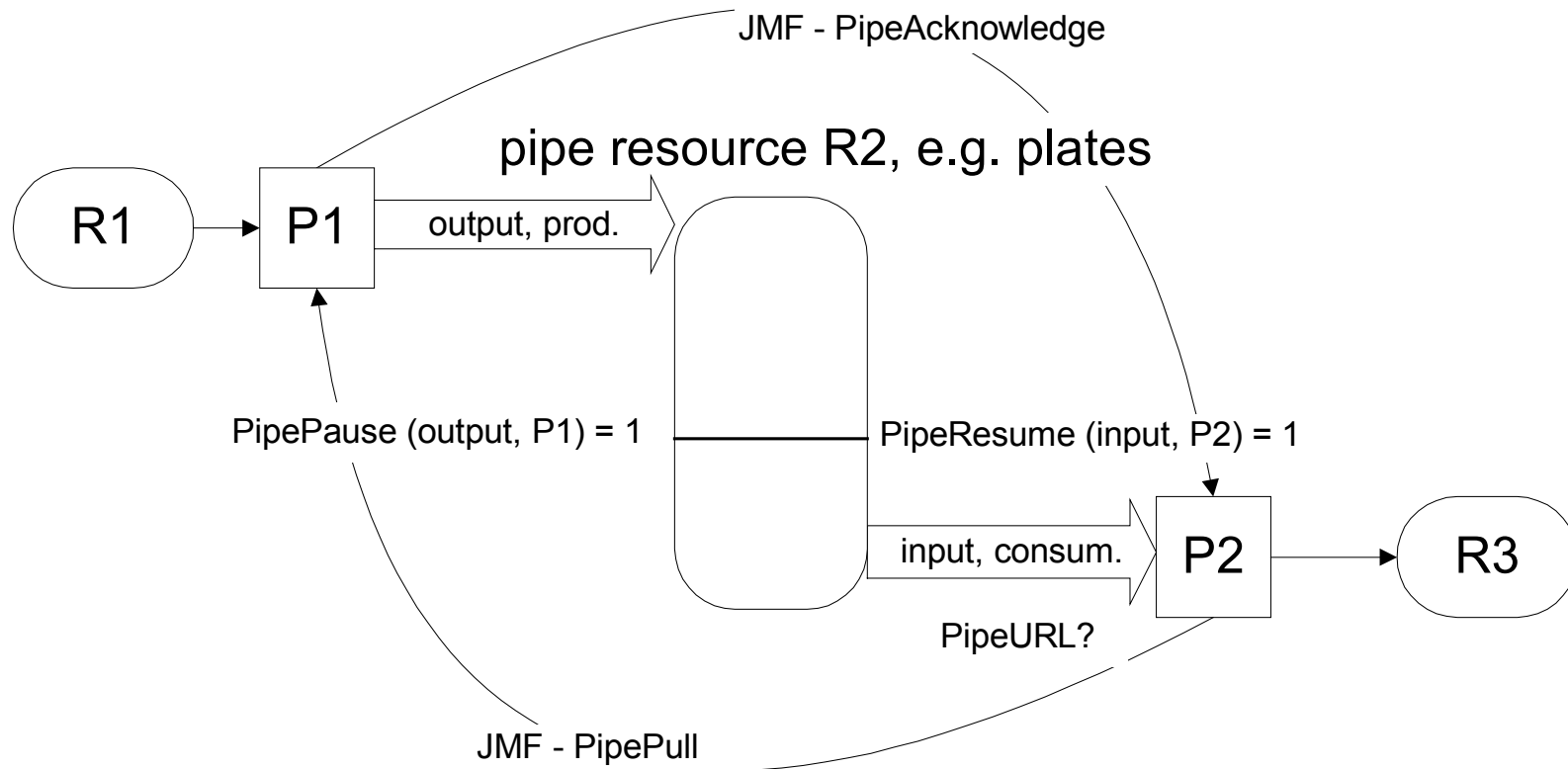
Pipes (Transient Resources)

- Overlapping Processing
 - Print 10 pallets and start folding when one is ready;
- Undefined Amounts
 - Request new plates in long press runs
- Data streams
- Buffer Handling
- Synchronization Messages

Pipe Resource Linking



Dynamic Pipe Linking (via JMF)





Audit Objects (I)

- Logging of Job Execution
 - Start time
 - End time
 - Phases
- Logging of Late Changes
 - Resources (used 85g Paper instead of 80g)
 - Consumables
- Status Summary
- Event Log

Customer / Node Information

- CustomerInfo
 - Map Subcontracting via Localized Customer Information in any JDF Node
 - Customer ID
 - Addresses (Delivery, Invoice, ...)
- NodeInfo
 - Scheduling
 - Deadlines
 - Processing Time Estimation



JMF Messaging Properties

- Dynamic Process Interaction
- Five Message Families
 - **Signal**: Unidirectional Post
 - **Query**: Request for information
 - **Command**: Request for a state change
 - **Response**: Immediate Answer to Query or Command
 - **Acknowledge**: Delayed Answer to Query or Command
- Message Protocol using HTTP
 - Firewalls are not an obstacle
 - Easy to use and implement



JMF Message Types (I)

- Initialization
 - Registration
 - Publish JDF Capabilities
- Device / Job Status / Progress Information
 - Consumable Level
 - Progress / Status
 - Settings
 - Currently executing jobs
 - Job Tracking



JMF Message Types (II)

- Queue Handling
 - Set Priority
 - Reorder / Group jobs
 - Hold / restart queued jobs (NOT running jobs)
 - Abort running Job
- Job Submission
 - Submission via HTTP
 - File Based JDF submission
 - hot folder
 - URL



JMF Message Types (III)

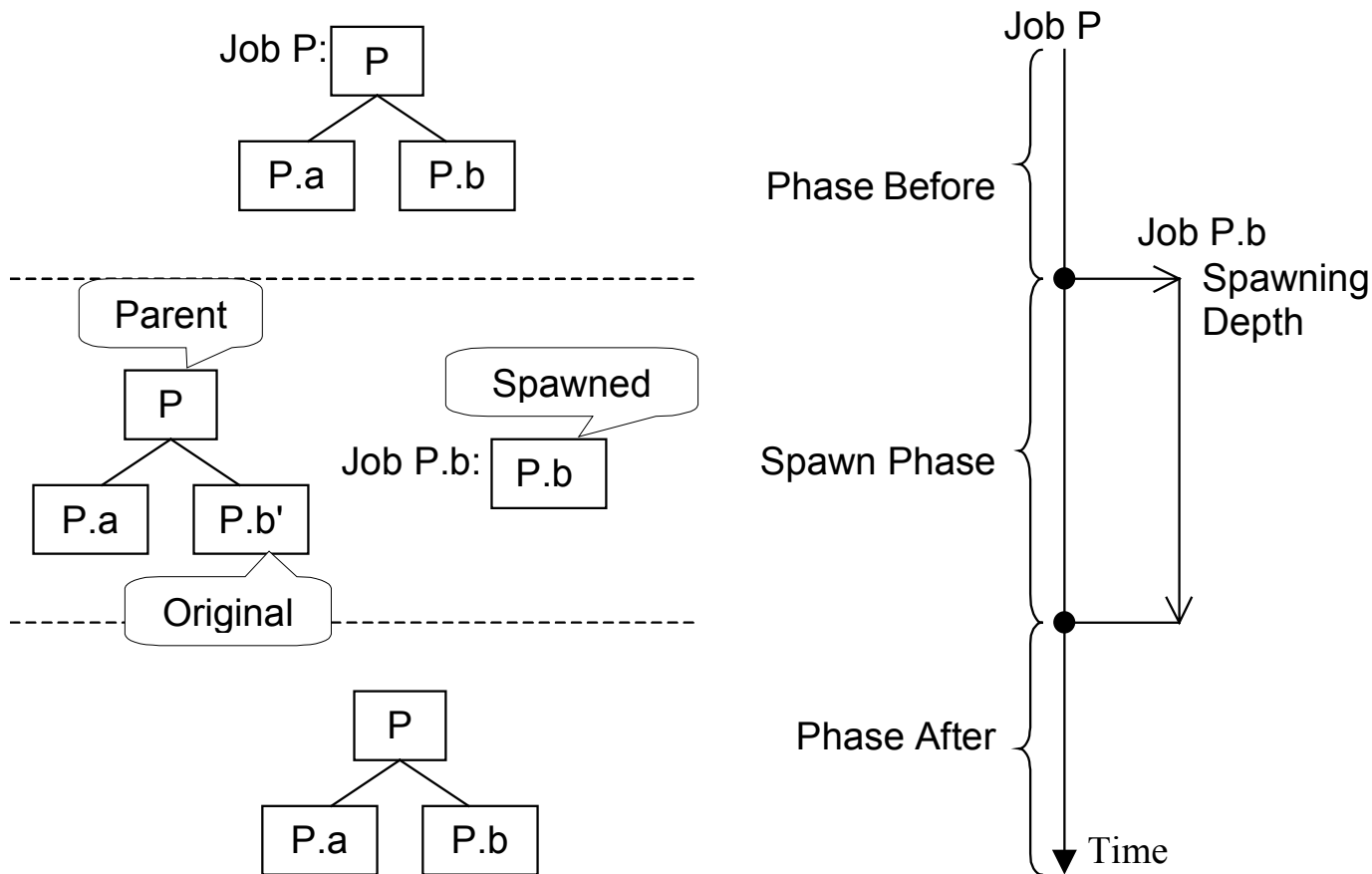
- Pipe Interactions
 - models the phone call from operator to operator
 - Start Production
 - Stop Production
 - High Water / Low Water marks
 - Individual Resource Requests
 - Resource Changes for iterative processing



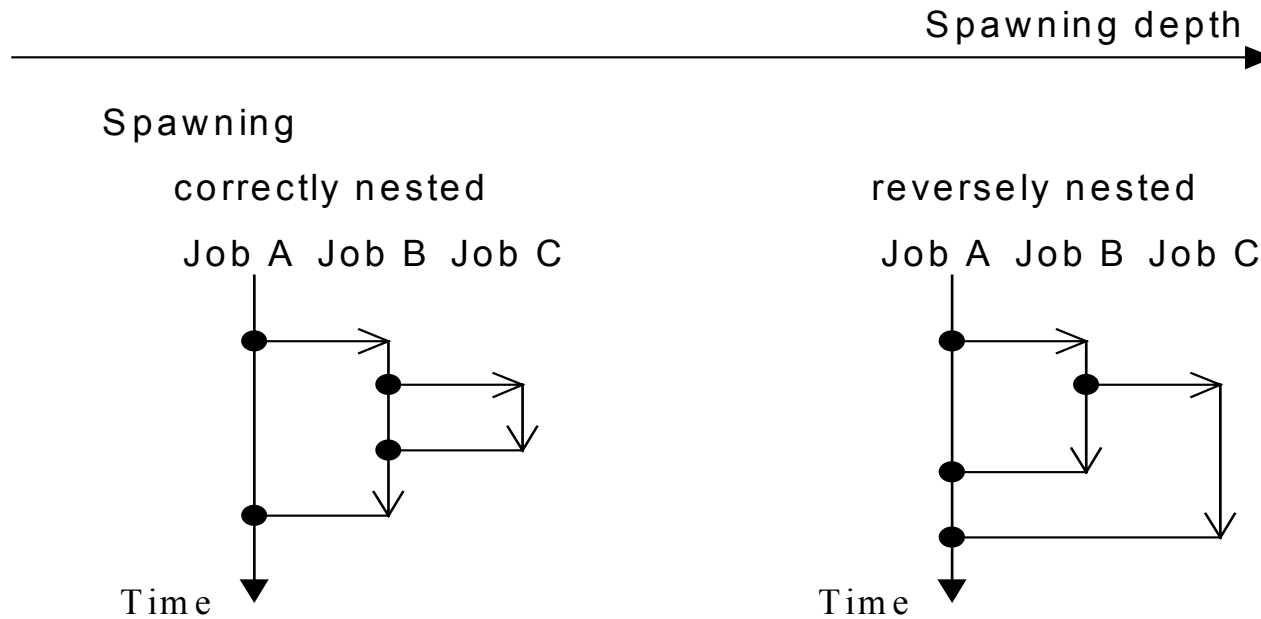
JDF - Spawning and Merging

- Spawn parts of the JDF Tree for independent Processing
- Merge back after Processing
 - Basic Mechanism
 - Recursive Spawning and Merging
 - Independent Spawning and Merging

Basic Spawning and Merging Mechanism

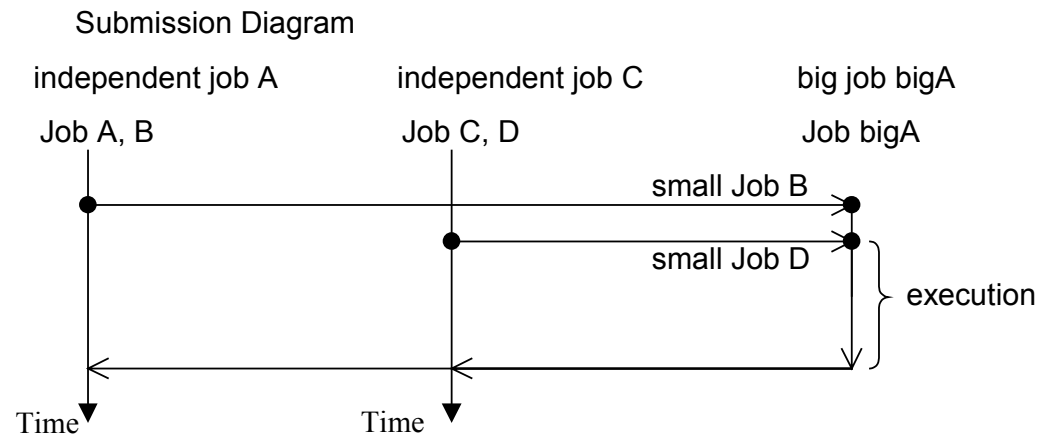
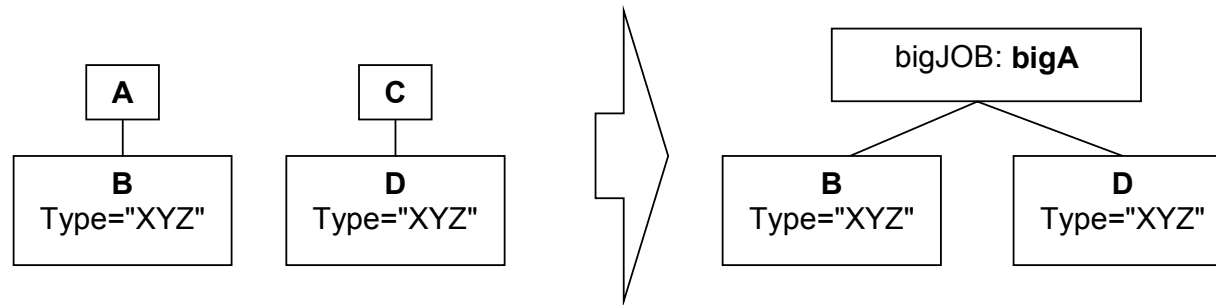


Recursive Spawning & Merging



Merging of Independent Jobs

e.g. for Combined Processing in one Press Run



Encoding Details

- XML Encoding
- External references via URI/URL
- XML Schema for Element Definition
- Extensibility using XML name spaces
- Optionally in a MIME/Multipart Wrapper
 - Allows single file with binary data
- Image Preview Data as Multiple PNG (Portable Network Graphics) Greyscale Separations

Extensibility Mechanisms

- Extensions are allowed in a private name space.
- Add attributes / elements to Resource

```
<Ink (...) HDM:Radiation="42" />
```

- Create New Resource

```
<HDM:JavaParams HDM:Milk="true" />
```

- Create new Process Type

```
<JDF (...) Type="HDM:JavaCooking" >
```

```
<ResourceLinkPool>
```

```
<HDM:JavaParamsLink Usage="input" />
```

PJTF Functionality

- File / Document Structure is replaced by a linear Run List
- Use Long Keywords from PJTF Specification
- Layout Tree (Sheet, Surface, PlacedObject...)
- Color description Objects
- Trapping Description
- Preflight description

CIP3 PPF (I)

- Press Instructions
 - Ink Zone Preset Values
- Finishing Instructions
- CIP3 Sheet Description
- CIP3 v3.1 Product descriptions
- CIP3 Web Printing Enhancements to be included

CIP3 PPF (II)

- CIP3 attributes are moved to their “Natural” positions within the JDF structure.
 - Sheet attributes are in the PJTF Sheet
 - Job Attributes are in the appropriate Node
- Preview Images are externally linked PNG separations
- Page Markup will be linked to its Process
 - Cut Marks <<==>> Cutting Procedure

IFRATrack

- JDF abstract product Model can be reduced to the IFRATrack Newspaper Model
- Job Tracking via Node and Resource Status
- JDF Messages are a superset of IFRATrack messages
- JDF LayoutElements confirm to IFRAtrack elements
- Deadline Tracking is defined



Variable Data Handling

- Provisions for Batch FileSpec Definition
- PPML and JDF are compatible
 - PPML defines:
 - variable document contents
 - Content Usage Statistics
 - JDF defines:
 - Production workflow for variable data
 - Interface to MIS
 - Imposition is overlapping but conflicts can be resolved.
 - JDF overrides PPML