

# PWG MFD Working Group Face-to-Face Meeting Minutes

At MWAI, Scottsdale, AZ

February 9-11, 2010

**Tuesday, February 9**

## 1. Attendees:

Nancy Chen, Oki Data  
Ira McDonald, High North (representing Samsung)  
Glen Petrie, Epson  
Jerry Thrasher, Lexmark  
Bill Wagner, TIC  
Rick Yardumian, Canon  
Peter Zehler, Xerox

## 2. Minutes Taker: Nancy Chen

## 3. Agenda:

### Tuesday

9:00-9:15 : Introductions, Assign Minute Taker(s)

9:15-10:30 : Detailed review of Overall MFD specification  
<<ftp://ftp.pwg.org/pub/pwg/mfd/wd/wd-mfdoverallmod10-20100203.pdf>>

10:30-10:45: Break

10:45-12:00: High level review of the FaxOut specification  
<<ftp://ftp.pwg.org/pub/pwg/mfd/wd/wd-mfdfaxoutmodel10-20100126.pdf>>

1:15-2:45: Detailed review of Copy Service specification  
<<ftp://ftp.pwg.org/pub/pwg/mfd/wd/wd-mfdcopymodel10-20100102.pdf>>

### Wednesday

9:00-9:40 : Discussion of goal for the MFD modeling effort  
<<ftp://ftp.pwg.org/pub/pwg/mfd/white/GoalForTheMFDModelingEffort20100114.pdf>>

9:40-10:20 : Discussion of MFD System Service

10:20-10:30: Next Steps

## 4. Detailed Review of Overall MFD specification

The followings are group's consensuses:

- Sec 2.6 Data Types – Ira will provide his revision as soon as he is freed up from his work load in other working Groups.
- Table 5 – Description of Common Subunit Elements
  - The PowerMonitor complex element – a sequence of MonitorEntry instances, of which constituent simple elements (i.e. ID, PowerState, PowerStateMessage) do not have data type defined in XML Schema yet.

- **AI:** Bill to change the type of ID from OID to integer type.
  - **AI:** Pete to add keyword definition of PowerStateWKV for PowerState enumerations in Schema. Note: In Power MIB, only primary power states (on, standby, suspend, hibernate, offsoft) can have suffix 1~5 for vendor extensions. Rest and Interrupt do not have vendor extensions.
  - In any subunit status, there is only one PowerMonitor group, which consists of PowerState and PowerStateMessage. PowerMonitor should not have multiple instances; should not have an ID (redundant with subunit ID).
    - **AI:** Pete to remove Id from PowerMonitor in all subunits.
    - **AI:** Pete to remove multiple instances from PowerMonitor.
    - **AI:** Pete to remove extension point from PowerMonitor.
    - PowerStateMessage – a String type, not extensible.
  - System has multiple instances of PowerLog which contains PowerMonitor and some other elements.
    - ID - power log instance ID & ComponentID - reference a Subunit ID
      - **AI:** Pete to rename ComponentID to ComponentReferenceID
  - **AI:** Pete to replicate SystemCapabilities (currently only consists of PowerSupport & Power Transition) under SubunitDescription; this is to exteniate/override what's in System when needed.
  - **AI:** Pete and Ira to make sure the three Power Policy groups, and Power Counter and Power Meter groups are in the System at least.
  - **AI:** Bill will update the Overall document with the agreed changes.
- Table 7 – Cover Elements
    - CoverStatus of which CoverSubunitType is not yet defined in Schema
    - AI:** Pete to fill out CoverSubunitType (cover, interlock, unknown)
      - CoverStatus – CoverSubunitType - <NONE> not resolved in schema
  - Table 17 – InputTrayStatus Elements
 

Confirmed the change:

    - FeedDirection Description – Added For example ShortEdgeFeed or LongEdgeFeed
  - Sec.4.4.3 JobProcessingCapabilities
    - Agreed to put forward reference to where it was described, no additional description here.
    - **AI:** Figure 47 – JobProcessingCapabilities needs to be updated
    - **AI:** Add reference - JobHoldUntil & JobHoldUntilTime were in IPP Production Print v2 (Job & Printer Set 2), section 5.4 in the old draft.
  - Sec 5.2.3.1 DelayOutputUntil & DelayOutputUntilTime – added new description
    - No objection.
    - **AI:** Ira to propose DelayOutputUntil & DelayOutputUntilTime to IPP JPS 2 semantics

**[Note]** At the time of publishing this minutes, Ira has already sent the request for adding DelayOutputUntil and DelayOutputUntilTime to IPP editors and got positive feedback.
  - Discussed the MFD operations that should be covered in the Overall Model and Semantics.

Agreed to Overall Operations	Type
Add<service>HardcopyDocument	user
Cancel<service>Document	User
Cancel<service>Job	User
Cancel<service>Jobs	Admin
CancelCurrent<service>Job	User
CancelMy<service>Jobs	User
	User
Create<service>Job	User
Disable<service>Service	Admin
Enable<service>Service	Admin
Get<service>DocumentElements	User
Get<service>Documents	User
Get<service>JobElements	User
Get<service>JobHistory	User
Get<service>ServiceElements	User
GetActive<service>Jobs,	User
Hold<service>Job	User
HoldNew<service>Jobs	Admin
Pause<service>Service	Admin
Pause<service>ServiceAfterCurrentJob	Admin
Promote<service>job	Admin
Release<service>Job	User
ReleaseNew<service>Jobs	Admin
Restart<service>Service	Admin
Resubmit<service>Job	User
Resume<service>Job	User

Resume<service>Service	User
Send<service>Document	User
Send<service>URI	User
Set<service>DocumentElements	User
Set<service>JobElements	User
Set<service>ServiceElements	User
Shutdown<service>Service	Admin
Startup<service>Service	Admin
SuspendCurrent<service>Job	User
Validate<service>DocumentTicket	User
Validate<service>JobTicket	User

- Close<service>Job operation currently is not in IPP or MFD Semantics. In IPP, Close-Job is implicit operation of a Boolean attribute when “Last-Document” set to “true” in Send-Document operation, making streaming very difficult. Ira will propose to IPP WG to add “Close-Job” operation to IPP JPS2.

[Note] In Wednesday face-to-face IPP WG session, Close-Job operation was added to IPP JPS2. Therefore, Close<service>Job will be added to MFD Overall Semantics in the next revision.

- Reprocess-Job and Restart-Job are print specific.
- Suspend-Job operation can only suspend a current job. Therefore SuspendCurrent<service>Job in MFD. If there are more than one current jobs, the system decide which job to suspend or suspend the optionally specified JobId.
- Sec 7.3 Service Operations
  - How detail the description of operations in table should be?
    - We agreed to reference IPP document only, no further descriptions required in the Overall semantics.
- Table 49, agreed that
  - Cancel<service>Jobs – admin operation
  - Cancel<service>Job - user
  - Cancel<service>Document - user
  - Hold<service>Job - user
  - Promote<service>Job - admin
  - HoldNew, SuspendNew - admin
  - Release<service>Job – user
  - Resume<service>Job - user

- SuspendCurrent<service>Job -user
  - Validate<service>DocumentTicket –user
  - Add 2 new ops from IPP
  - Table 52 – add RemoveCurrentJobs – admin
  - CancelCurrentJob – user
- Promote<service>Job – the new text was reviewed. Agreed that
    - This operation has an optional JobId as parameter in IPP. The JobId is mandatory in MFD, and PredecessorJob is optional and if specified, the job promoted is to be scheduled after the specified predecessor job. [Ref-3898] – the job is scheduled after the current job or the specified predecessor job. If predecessor job is not found, (note: must be in processingStopped, pending, or processing state), service must reject and return “processing the job is not possible”.
  - SuspendCurrent<>Job – the new text was reviewed. Agreed that
    - On Line 2013-2014 simply take out the entire statement : “The printer must reject...”
    - One Line 2011-2012, add qualification to “If more than ..., all jobs must be suspended.” with an exception that if a JobId is specified, “only the job identified by the specified JobId must be suspended”.
    - Agreed with last ResumeJob statements on lines 2015-2016.

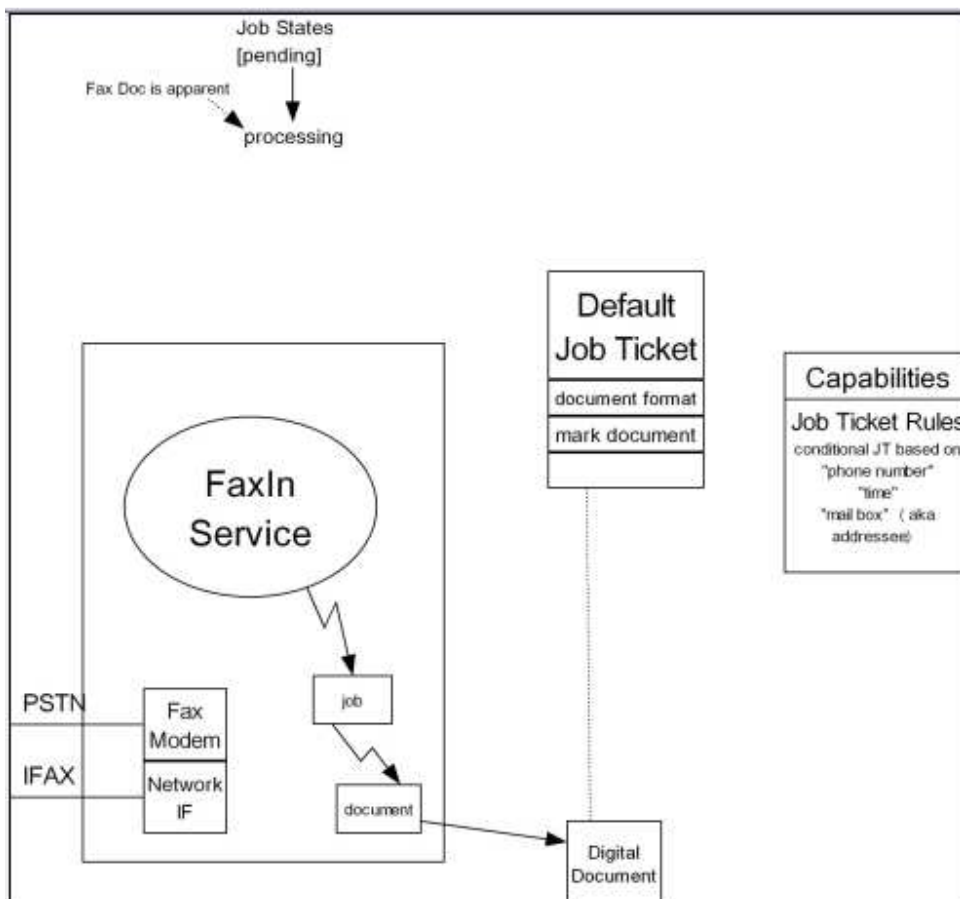
## 5. FaxOut Specification High Level Review

- Will discuss highlevel FaxIn this afternoon
- White paper tomorrow
- Model Overview –
  - Use the same XML Schema for all MFD services; only that resource service does not have JobTable.
  - FaxOutServiceCapabilitiesReady – models what’s loaded on the device, ready for job processing. FaxOutServiceCapabilities models what can be supported by the device.
  - **AI:** For legal compliance, it requires that jobs need to be **durably** logged before removed from Job History. => Add “durably”.
  - **AI:** MultipleOperationTimeout between two operations (e.g. Create-Job, Send-Document ), MultipleOperationTimeOutAction, and MultipleDeocumentJobSupported should be in MFD Overall Semantics spec.
  - **AI:** Fax Log URI – should be **Durable** Log URI in description.
  - For the need of tracking number of documents flow through a MFD, there is a counter for number of documents in each job. A vendor can extend the model with a counter for the total number of documents by adding up number of documents in each job.
  - HeaderPrint has three enumeration keywords: Inside(on the impression sheet), Outside, and NoPrint
  - FaxOut has multiple destinations. DestinationStatus under JobStatus records which destination fax has succeeded.
  - FaxModem subunit is defined in MFD Overall Semantics spec.

- Get<Document/Job/Service>Elements – right now returns high level group elements. If this element is omitted in the request message, MUST all groups are returned?
  - Agreed that NO. The default is to return only the identifying information for that object in general, i.e. Job ID, Document ID, or Service ID.
- In the tables of interface operations, bold font implies mandatory attributes, the rest are optional.
- **AI:** in Conformance section, change CopyService , Scan Service to FaxOut Service.

## 6. FaxIn Service High Level Discussion

- The diagram below is a straw-man high level FaxIn Service model discussed in the meeting:



- FaxIn Service use case scenario:
  - Fax data comes in through PSTN line or IFax network line received by Fax Board Interface (modem/network interface).
  - FaxIn job is created using a DefaultJobTicket, after Fax negotiation completed, then starts to receive data. (Note: this way “calling a wrong number” will not cause creation of a fax job.)
  - Associated with the job, a document object is created in which FaxIn data is spooled.

- FaxIn job proceeds from Pending (Job Incoming state) to Job Processing State (Fax document is arriving)
- FaxIn Service has a set of defaultJobTickets, each has JobTicketRules based on which the user selects the type of FaxIn job to be accepted.
- Job Ticket attributes contains:
  - Document Format for digital document to be outputted/stored in a repository
  - Marked Document – to indicate the digital document is to be outputted to marker. (Note that with this attribute, a single function FaxIn device can be provided without needing to implement the full Print Service functionality.)
  - Conditional Job Termination based on “mail box” (subaddressing), “phone number”, “Time”, “fax protocol”, ..., etc.
- FaxInServiceCapabilities contains JobProcessing and DocumentProcessing capabilities that restrict the range of values Job Ticket attributes for job processing and document processing can have.
- JobTicketRules is a sequence of conditions sparsely populated in Job Ticket that can be merged with the DefaultJobTicket to determine how an incoming FaxIn job should be terminated.
- Job Status is continuously being updated by automata while job is being processed.
- When the digital document is closed, the document processing is completed.
- When the digital document is outputted, the job processing is completed.
- Ira has volunteered to write the FaxIn Service specification.

## 7. MFD System Service Operational Interface Review

- **AI:** Global change for MFD Service to System Service
- Agreed that System Service should have the following operational services:
  - Disable/Enable all services
  - Pause/Resume all services
  - Restart/Shutdown an individual service by Service ID and Service Type
  - Startup an individual service with optional Service ID, if ID already taken, the operation failed. This is for remote system management, using all default configuration and values for attributes.
  - GetSystemElements – to replace SNMP, enable retrieval of any element in the MFD model at System level. Simple name of the element in the request message is not enough, because the individual service object needs to be identified. Should Xpath be used instead?
    - Simple Xpath expression in order to top level and one level down elements should be suffice.
    - Want to be able to only get the counters in SystemStatus
  - SetSystemElements -
    - For Set, we want to be able to exclude some elements in SystemStatus which should be set by automata. Use sparsely populated tree for Set elements.

- Subunit elements should only be set at the system level. For setting elements in subunits, need to be able to get to deep level of elements, and should include subunit ID as the key element to identify the target subunit to be set. Subunit Type should never be set.
    - **AI:** Setting SystemConfiguration does not need a subunit ID for the set operation.
      - **AI:** add RestartAllServices, DeleteService for an individual service.
- There is an initial MFD System specification file “MFDSystemAnd AdministrationService” draft posted on PWG MFD ftp web site.

**8. Copy Service Specification** was not discussed due to none has read the document.

## Wednesday, February 10

### 1. Attendees:

Nancy Chen, Oki Data  
 Ira McDonald, High North (representing Samsung)  
 Joe Murdock, Sharp Lab America  
 Glen Petrie, Epson  
 Jerry Thrasher, Lexmark  
 Bill Wagner, TIC  
 Peter Zehler, Xerox

### 2. Minutes Taker: Nancy Chen

### 3. Agenda:

- Discuss the whitepaper: “Goal of Multifunction Device Modeling Effort”.
- Review the Copy Service Specification when there is time left.

### 4. Discussion of the Goal of Multifunction Device Modeling Effort Whitepaper

- Pete went through the whitepaper section by section with the attendees; the following points were emphasized:
  - The evolution of the PWG Semantic Models v1.0, through v1.1 and now v2.0 (MFD Semantic Model) in progress has built consensus on a common printer/MFD model that has benefited device and service vendors in rapid development of new products and applications using various protocol bindings with consistent responses. The common model has made possible a thin Protocol Gateway being built, providing the syntactic translation required by multiple protocol implementations (e.g., IPP, WS-Print, JDF-Digital Print, SNMP, CIM), thus resulted in reduced product development cost, increased reliability, and quicker time to market for print service related products. The common set of MFD semantics can be applied to different environments and applications to



- allow environment specific solutions, including service advertisement, discovery, monitoring, management, job submission and tracking, facilitate workflow solutions.
- The model is extensible, yet still based on a common semantic model, allows scaling of the model to different product implementations.
  - There are many benefits for a web services binding of the common MFD semantics:
    - XML data binding does not restrict to only data types defined in rfc2911: e.g. floating Point (Xerox needs). There are a lot more tool sets can be used comparing to binary encoding that's only PWG-specific!
    - Allow the use of tools to generate code for client and server; nothing is available for IPP.
    - WS-Discovery: provides for the discovery of devices and its hosted services in an ad hoc environment; can be used in concert with other discovery methods (e.g. UDDI, WS-Discovery Remote extensions, Bonjour) for discovery in managed environments, enterprises, or the Internet.
    - WS-Transfer: quite useful, gives information on particular devices, including URL/service types/deviceID/other metadata, very useful for WS-Discovery.
    - WS-Eventing: provide very lightweight eventing via SOAP over UDP, TCP not required.
    - WS-Addressing: provide protocol independent device address via standard GUID, which is not lost over power cycle.
    - DPWS – provides a set of requirements that a device must meet for supporting web services.
    - WSBEPL – a workflow language that can integrate MFD web services into office workflow.
  - One major barrier for MFD interface protocol is the integration into OS environments. With the interface standard for writing office workflow solution using MFD as on/off ramp devices, device does not need to know the specific interfaces in the workflow. In the meanwhile device vendors can implement BEPL on device for integration with an office workflow.
  - The abstract PWG Semantic Model v1 is realized in its concrete mapping to IPP, UPnP Basic Print, Java Print APIs, WS-Print, LDAP Printer Schema, SLP Printer Template, etc. Similarly the PWG Semantic Model v2 will be realized in the mapping of all MFD imaging services to concrete protocols. Vendors' differentiation advantages come best from being able to leverage vendor-specific extensions to the standard, with well supported capabilities, giving their products an edge in the target marketplace.
  - Since MFD semantics not only is consistent with IPP in Print Service, but also expand IPP to include other MFD services. web services can be implemented by building a syntactic gateway very rapidly on top of any other protocol, IMFP (Internet Multifunction Protocol) Everywhere can easily be a follow-on of IPP Everywhere. In the meanwhile, IPP Everywhere standardizes on limited set of supported document formats, can benefits from MFD Transform Service to convert document from one format to another.

## **5. Status and Remaining Work**

- Completed documents: Print Service, Scan Service, Resource Service specifications are completed.
- MFD Model and Overall Semantics, MFD System Service, Copy Service and FaxOut Service specifications are currently under development. Copy Service is near completion, soon in Prototype draft status.
- Remaining work:
  - EmailOut and Transform services are relatively straightforward
  - EmailIn and FaxIn services require more thoughts

## **6. Proposed Timeline**

- Q3 2010 to complete all service specifications currently under development
- Q4 2010 to complete the remaining specifications

## **7. Next Steps**

- Next MFD concall on Feb 25, 2010, Thursday, 3pm EST.