Meeting Minutes
PWG MFD Working Group Face-to-Face Meeting
Sharp Labs, Camas, WA
August 13-15, 2008

On-Site Attendees:

Nam Heo	Samsung
Ron Bergman	Ricoh
Nancy Chen	Oki Data
Lee Farrell	Canon
Mike Fenelon	Microsoft
Joe Murdock	Sharp Lab
Harry Lewis	InfoPrint Solutions
Glen Petrie	Epson
Jerry Thrasher	Lexmark
Bill Wagner	Konica Minolta
David Whitehead	Lexmark
Craig Whittle	Sharp Lab
Peter Zehler	Xerox

Phone-In Participants:

Shah Bhatti	Samsung

August 14, 2008 Meeting

1. Meeting Agenda

2:00-2:15 : Introductions, Assign Minute Taker(s)
2:15-2:30 : Current status and future schedule/milestones
2:30-3:00: Discuss results of Requirements document Formal Vote, and initiating Last Call of Network Scan Service Model and Interface specification
3:00-3:30: Begin review of Resource Service specification starting with PWG model conflicts and high level issues
3:30-3:45 : Break
3:45-5:30: Continue Review

*No adjustment made to the agenda

2. Minutes Takers Assigned – Nancy Chen

3. Current Status and Future Schedule/Milestones

- This meeting - final review of the Scan Service specification,
  preparing for the PWG Last Call for comments,
  continue the review of Resource Service draft.
  The Scan Service Use Cases and Requirement specification has passed the PWG Last Call for comments. Harry Lewis will provide the final tally announcement at the MFD WG email list.
- Next meeting - complete the Last Call of the Scan Service specification,
  final review of the Resource Service specification.
- Next year - start to specify other services.

4. Review of Scan Service Specification

- **Question on whether the "pull" model of scan documents used by WS-Scan should be incorporated in the PWG Scan Service.**

  The main concern was that a product that has already implemented WS-Scan might not comply with the PWG Scan Service model which always "pushes" scan documents to URI destinations. It was recognized that in WS-Scan, scan document destination is a part of WS-Eventing binding; once the client is notified that scan document is available, the client must then come to retrieve the document (called device initiated scanning). The same binding can be used by the PWG Scan Service; once the scan document is pushed to its "storage" destination URI, the storage can notify the client which then pulls the document. There is no "clash" or exclusion of the WS-Scan's "pull" model. But it does require implementing WS-Eventing or a Storage Service to allow the "pull" from a client. WS-Scan is a web service protocol binding of the PWG Scan Service model utilizing WS-Eventing. Its 'pull" data model does need to worry about a "wrong" client has been notified to pull the data. The PWG Scan Service can be mapped to any binding protocol, providing wider interoperability, and better security since the data can be pushed to a central location that can provide policy enforced access control. In the future it might be a good idea for PWG to define a Storage Service that provide interoperable "pulling" of document data from all MFD services.

  Conclusion: No action required.

- **Peter Zehler initiated the PWG Last Call for Scan Service specification.** There was no objection from the members. Bill Wagner's editorial comments received today will be incorporated into the next updates before the formal Last Call announcement.

5. Review of Resource Service Specification

- **PWG Model conflicts and High Level Issues**
  1. In the previous PWG model, "Resources" are various components that installed on a MFD system. In Resource Service a "Resource" is an object required for performing a task. This two "Resources" conflict with each other because they actual means different things. The "Resource" in the previous PWG model is now named "ResidentResource" which is a firmware, software, logo, image, font, or form that are installed on a MFD. ResidentResources are permanent resources for a specific MFD, not meant to be used as a general retrievable resource by the Resource Service, whereas the "Resources" served by Resource Service can be stored and retrieved by any client that is authorized with the

appropriate access rights. The type of Resources in the scope of Resource Services are font, form, image, logo, ICC profile, template, whereas firmware and software are executables out of scope.

We had a lengthy discussion on whether the newly adopted ResidentResource is an appropriate name since it includes firmware and software which are out of scope by the new resource definition for Resource Service. The reason for ResidentResource is so that they are instantly available for use.

We concluded with changing "ResidentResource" to "AvailableResource" and classifying the resources into two categories:

* Static Resources : non-executable resources, that include font, form, image, logo, ICC Profile, template, are servable by Resource Service.
* Executable Resources include firmware and software. Both static resources and Executable resources can be installed in a specific MFD system as Available Resources, but not to be used by the Resource Service.

A client can use the Resource Service to retrieve the data of a resource and install it in a MFD. Once installed, the client can only use a MFD service to query the metadata of an installed "AvailableResource", not retrieve the installed resource data.

2. Review Resource Service Specification

- Remove lines 179-180: resource might not expensive.
- The resources defined here are resources specific to MFD.
- Remove "Global Resource", "Resident Resource", "Document Resource", "Job Resource", "Scan Job Resource", Add "Resource" definition.
- Remove lines 245-246 that include software, firmware as resource type.
- Section 6.1.1.2 – expand the PWG resources into two categories: Static type and Executable type resources. The Resource Service will have one more parameter to allow filtering on one resource list into static or executable resources by Get/List Resource operations. The PWG model will maintain two Available Resource lists – one for static resources, one for executable resources – this is much easier for MFD service operations such as GetScanElement to get different categories of Available Resources.
- "Software" and "Firmware" are executable resources – change definition.
- Section 3.3.2 – revise, remove "firmware", "software" from out of scope resources.
- Section 4 Figure 1 – The relationship between the remote client and local Resource Service should be many-to-many association.
- Line 236 change destroy to "delete".
- Section 5 – remove lines 248-251: now definition of resources has been changed. Available resources might have restrictions on their distribution by certain policy.
- Line 258 – A resource is not available until the operations on the resource has completed.
- Section 6 – "Storages" subunit needs to be added into schema.
- Section 6.2 – Resource Service Description:

        - Although may not adequate to be in this group, we do need a "NaturalLanguageSupported" element in order to support "NaturalLanguage" element in Resource Service Status for system generated state messages.
            - Pete will add a "ResourceServiceCapabilities" element that contains only "NaturalLanguageSupported" element.
        - The Sequence of ##Other element and the ##Other attribute will be expanded into all Resource Service group element so that vendors are not only allowed to extend a group element, and also extend attributes of an element.

August 15, 2008 Meeting

1. Meeting Agenda:

2:00-2:30 : Discuss next MFD service to address
2:30-3:15 : Continue Review of Resource Service
3:15-3:30 : Break
3:30-4:45 : Continue Resource Service discussion
4:45-5:00 : Next Steps

2. Discussion of Next MFD Service to Address

- We tried to identify the next core service like print, scan, transform, storage so that other complex services or workflows can be built on the top. The core services are those complex enough for which external interfaces exposed can be used for remote job submission; and workflow jobs with underlying sub-jobs flowing from one core service to another can be tracked or rolled back when necessary. We do not intend to define the basic services for composing a MFD service – for example, copy is a core service, but it's not composed from scan and print service. Copy service is not interesting because it's mainly for providing local operations. The list of preferred other core services named by member are: EmailIn, EmailOut, FaxIn, FaxOut, NetFaxIn, NetFaxOut, Transform, Storage.
- Transform and Storage are lower level core services that can be used by other services to, for example, transform/store any data to connect MFD workflow from one service to another. However in order to understand all services they need to provide, other services using them need to be defined.
- Email and Fax services are equally favored as services that can probably provide useful applications. However, nobody knows for sure the functions of EmailIn and EmailOut services. FaxOut is useful for remotely printing any document to a MFD which then send it out as a PSTN fax. Standardizing FaxOut Service allows interoperability of remote fax job submissions. FaxOut was agreed for the next MFD core service to be defined also because we have an editor and fax experts for the service.

3. Continue the Review of Resource Service Specification

- Section 6.3
    1. Serial Number – this is an optional element only applicable when the Resource Service is hosted on MFD, thereby you can publish the MFD serial number.

2. Why condition table – This table captures all subunit alerts, such as storage offline condition, etc. – that affect the ability to retrieve or store resources.
3. Currently there is no administrative operation in Resource Service – Add to schema Enable, Disable, Restart, and Shutdown operations.
4. Remove “InternalStorageFull” : “StorageFull” and “StorageAlmostFull” are sufficient to represent the condition for all type of “Storages” subunits such as internal/external storage, flash, RAM, …, etc. Each condition has a pointer to reference the subunit causing the condition.
5. We will continue to use ResourceID as the key for a created and stored resource, and for GetResourceElement, DeleteResource, etc. Using ResourceDataURI is not as secure as ResourceID.
6. Are there additional descriptive information required for a self-describing resource? Now we have two categories of resources, for example, the client platform name where an executable resource will run? This is good information for querying what executable resources are available for a particular platform.

   AI: ALL - We need to investigate further additional metadata required for filtering resources. We need further investigation on better resource type name for static and executable resources.

   Ideas for other descriptive elements of a resource:
   - creation date
   - Originating user
   - protection code (access mode)
   - resource type
   - MIME type (but what MIME type is for font?)
   - Namespace
   - Target environment for executables

   Please send your idea/investigation results whenever you have them.

7. Need to add ICC profile to resource type.
8. ResourceStatus contains only ResourceID. Should we add some other elements? For example “number of access” - a counter, “access method”?
   There was no strong request from members.
9. Section 7 Theory of Operation –
   - Remove “UpdateResource”, line 560

10. Section 7.1 Service Operations
    Recommended changes –
    - Remove Update Resource with ReplaceResource, and others in Operation concept
    - Change GetResource to GetResourcElement, but return ResourceID not URI. Also we need to add resource type and resource category as filter.
    - Change PutResource to CreateResource, should return the self-descriptive information of the resource that will be defined later.

- Add SetResourceElement – this set the metadata/descriptive elements of the resource.