Printer Working Group

Semantic Model Work Group
Meeting Minutes
April 3, 2003
Teleconference replacing
Washington D.C meeting

1. Working Group Items

- The mailing list is active (sm@pwg.org)
- The Semantic Model web page is available. (http://www.pwg.org/sm)
- Weekly teleconferences are being held. (See mail list for details)

2. Semantic Model Document review

(Pete Zehler will make the changes and publish a new version of the document)

- Update with latest changes from Document Object specification.

3. Schemas review

(Pete Zehler will make the changes and publish a new version(s), Dave Hall and Pete Zehler will investigate tool compatibility, Bob Taylor and Dave Hall will investigate a loosely typed extension mechanism)

- The schema will be updated to reflect the changes in the Semantic Model.
- The conversation focused on support of extensibility mechanism by existing and emerging tools. At the core of the PWG schema is the ability for vendor extensibility
    - Object (i.e. element) Extensibility: This uses the #any construct in element definitions to permit any new element to be added to the object.
    - Keyword Value (i.e. NMTOKEN) Extensibility:. The preferred method defines the keyword as an element of type union containing the well-known values and extension pattern. Both of these would be of the same base type but with differing restriction (enumeration and pattern respectively). The current approach uses appinfo to reference the well-known values and extension patterns for the element
- Investigations will continue into tool set compatibility issues..
    - Some code generating tools had a problem with the union construct for keyword value extensibility. XML parsers seemed to handle this construct correctly.
    - Some prerelease parsers have trouble with the object extensibility construct. The #any is the standard and preferred method for element extensibility.
- The desire is to have two versions of the schema that are on the wire compatible.
    - We need to investigate to determine just what this means and if it is possible.
    - Version 0.93 will be changed back to the preferred way to model extensible keywords. The preferred method uses the union construct. Version 0.93 already uses the #any construct for object (i.e. element) extensibility.
    - A new version of the schema (ssm for simplified semantic model) will be created to try and have a tool friendly, on the wire compatible schema.
- Investigate a loosely typed mechanism that can give us the semantic equivalence of #any

- The proposal is an unbounded element sequence that contains name, value and type strings.
- Make sure mechanism supports extending elements with complex elements
- Make sure mechanism is compatible with tools

3. Semantic Model Process

- Not discussed.

4. Document Object

(Tom Hastings will make the changes and publish a new version to prepare for a page-turner review on Thursday April 10)

Resolutions to issues in Tom Hasting Mail notes

- (1) Close-Job operation needs the same timeout Printer requirements as RFC 2911 has for Send-Document.
  - Agreed.
- (2) Remove "document-id-uri" Document Description attribute. PSI agreed last week that they can use the IPP "document-number" (integer(1:MAX)) Document Description attribute to identify documents within a job.
  - Agreed.
- (3) Add a "document-format-version" (text(127)) operation/Document Description attribute with self-identifing values, such as 'PS/3', 'PCL/5e', 'PDF/1.4', 'PDF/X-1a:2001' (See ISSUES 01 and 02 below for more details). The prefix is from the Printer MIB langTC values used in the prtInterpretedLangFamily object and the version after the "/" can be used in the Printer MIB prtInterpreterLangLevel object.
  - Need an auto-sense value. The langAutomatic is in the Printer MIB, so we'll used 'Automatic' as the value for the "document-format-version" to mean auto-sense the version.
    If the client supplies "document-format-version", the client MUST also supply the "document-format" operation attribute. So the Printer isn't forced to parse the version.
    A client MAY supply "document-format-version" values from the Printer MIB that don't have their own MIME type, such as NPDL. In this case, the client MUST supply:
    "document-format" = 'application/octet-stream' and
    and the client MAY supply:
    "document-format-version" = 'NPDL'.
    If the format doesn't have a MIME type, but does have a version, then the "document-format-version" = "Xxxx/yyy', where Xxxx is from the langXxxx Printer MIB Textual convention with the 'lang' removed and yyy is the version.
    If the version is from a standard, then the standards body's designation for the standard is used, including part numbers, if applicable, and year. For example, for SPDL:
    the "document-format" = 'application/octet-stream' and
    the "document-format-version" = 'SPCL/ISO/IEC 10180:1995'.
- (4) Add a "document-format-version-default" (text(127)) and "document-format-version-supported" (1setOf text(127)) Printer Description attributes to describe the default and supported values. (See ISSUES 03 below for more details). So "document-format-version" and "document-format" would have parallel semantics, including also being member attributes of "document-format-details" (1setOf collection).
  - Agreed.

- (5) ISSUE 04 (repeat of below): Is "document-format-version" a Best Effort (hint) "job-mandatory-attributes" = 'document-format-version' or do we make "document-format-version" be like "document-format" in that the Printer MUST reject the job if the Printer doesn't support the version?
    - Agreed: We won't generalize "ipp-attribute-fidelity' to apply to operation attributes. Agreed: "document-format-version" will remain a hint. Same as for "digital-signature". However, if the Printer supports the "job-mandatory-attributes" attribute and the client supplies the "job-mandatory-attribute" operation attribute with the 'document-format-version' keyword value, then the Printer MUST reject the job if the "document-format-version" attribute is supplied with a value that isn't supported by the Printer (or the Printer doesn't support the "document-format-version" attribute at all).

- (6a) Put the version strings into a flat text file that implementers and implementations (at CD writing time, install time, vendor update time, at power-up time, etc.) can access on the PWG FTP site. Adding new values is simply done whenever they are registered or standardized with some recognized body, such as IANA, CIP4, ISO, etc. Also part of the Schema. Maintainer sends new proposed values to the PWG list for review.
  ISSUE 05: We need to convince CIP4 to use the same flat file for their FileSpec/@MimeOrFileTypeVersion attribute and/or have each of them shadow the other.
    - ACTION ITEM (Tom): Try to convince CIP4 to let PWG host the flat file, but that it contains values needed by CIP4 and PWG.
- (6b) The format of the file needs work so that it can be easily parsed. It won't be in XML. Something along the lines of a four column file, where each column is separated by something easy to parse. (Note: CIP4 FileType attribute can include white space "document-format", CIP4 FileType, "document-format-version", Reference
    - ACTION ITEM (Tom and Ira): Propose a format.
- (7a) Add a "document-natural-language" (naturalLanguage) operation/Document Description attribute and a "document-natural-language-supported" (1setOf naturalLanguage) Printer Description attribute. The "document-natural-language" operation attribute is already defined in RFC 2911 for use with Print-Job, Send-Document, and Send-URI, so we are just continuing this RFC 2911 attribute for the Create-Document operation and making it a Document Description attribute as well, so that it can be queried. Also keep "document-natural-language" (naturalLanguage) as a member attribute of "document-format-details" for describing packages.
  The "document-natural-language" is needed in order to know how to display characters that depend on language. Another hint at the top level, same a "digital-signature" and "document-format-version". But the client MAY supply "job-mandatory-attributes" with a 'document-natural-language' keyword value.
  ISSUE 06: What about multiple languages in a single document for the top level attribute?
    - Agreed: Keep single-valued at the top level. The "document-format-supported" Printer Description attribute indicates the values that are supported in the union of the document format supported. Note: Printer MAY include the region, e.g., en-US, en-GB, or may just have en.
- (7b) ISSUE 06a: What about for the member attribute of "document-format-details"?
    - AGREED (again): Make 1setOf for the member attribute.
- (8) Add a "document-charset" (charset) operation/Document Description attribute and a "docuemnt-charset-supported" (1setOf charset) Printer Descrition attribute. This attribute is needed for the many plain text and markup languages in which the charset is not embedded in the data. For example, PCL often doesn't have the charset escape sequence in the data. Also many files use the various 8-bit ISO 8859 charsets in which the lower half is US-ASCII and the

upper half is various Latin sets (about 8 or 9), Greek, Cyrllic, Hebrew, and Arabic. Shift JIS is another example where the left half is US-ASCII, but the right half can be one of a number of things. But if the data doesn't contain the charset escape sequences, this attribute can help the Printer know what the charset is in the Document.

AGREED: Keep single valued as in RFC 2911. It's not a hint. The Printer MUST reject the job if the supplied charset is not supported. Printer MAY also check the content while processing to see that the document data meets the coding requirements for the supplied charset. If the data contains something that is outside the charset encoding, the Printer MUST either (1) hold the job or (2) abort, and SHOULD NOT continue printing square boxes for encoded characters are outside the charset supplied by the client.

- (9) There is one issue embedded in the Document Object spec: 6.1.2.2 document-creator-application-version (text(127)) This OPTIONAL member Operation attribute identifies the version number of the application that created the document. The intent of this attribute is for display to a human being, rather than being parsed by the Printer for purpose of affecting the interpreting by the Printer and so may also include the name of the application, as well as build or service pack numbers. Examples: "Winzip 8.1 (4331)", "Acrobat 5.0.5 10/26/2001", "Microsoft Word 2000 (9.0.4119 SR-1)"
  ISSUE: OK that the purpose is human consumption, instead of program consumption and that it be the same as the application shows in its help message?
  - AGREED: OK. In "document-creator-application-version-implemented" versions: Its OK for the Printer to omit some of the suffix stuff, since it is for Human consumption. So for example, the Printer MAY return "Winzip 8.1", or "Microsoft Word 2000". Need a value of 'other' and 'any' Printer will try on other versions than in its supported list, unless the client supplied "job-mandatory-attributes" = 'document-format-details.document-creator-application-version'. The two other version attribute: "document-format-version" and "document-format-os-version" are intended for program consumption, not human consumption. Its possible that CIP4 will want to have both types of versions for: applications, operating systems, and documet-formats.
- At today's IPPFAX telecon, we discussed the following IPPFAX Printer Description attribute: digital-signatures-supported (1setOf type2 keyword) This attribute identifies which digital signatures technologies are supported by the Receiver. A Receiver MAY support this Printer Description attribute. (TODO: Get list of keywords; can be found in "IPP driver install" spec) The IPPFAX WG agreed: that it should go in the IPP Document object spec, and that it should have a "digital-signature" (type2 keyword) operation/Document Description attribute that the client submits in a Document Creation operation as well. And therefore, the spelling of the corresponding "digital-signature-supported" (1setOf type2 keyword) Printer Description attribute should be without the "s". The description from the "IPP Driver Install" (IPP Printer Installation Extension) spec is: "digital-signature" For IPP: One OPTIONAL LOWER-CASE 'keyword' string identifying the mechanism used to ensure the integrity and authenticity of this set of Client Print Support Files. Standard values are: 'smime', 'pgp', 'dss', and 'xmldsig' which are defined in [RFC2634], [RFC1991], [dss], and [xmldsig], respectively. In addition, the special keyword value: 'none' is valid.
  - If the Printer supports this attribute: The Printer MUST verify the signature according to the rule for that signature format. Different PDLs have different ways of embedding the signature. If the Printer supports the "document-signature" attribute, the Printer MUST accept the document and MAY either (1) ignore the signature or (2) put the job on hold. So if the "document-format" is supported, the Printer MAY accept the Job even if is doesn't understand the signature, and the Printer MUST return the attribute as unsupported in the Unsupported attributes group as in [RFC2911]. This Printer

behavior is backwards compatible with a Printer that doesn't support the "digital-signature" attribute. However, if the Printer supports the "job-mandatory-attributes" attribute and the client supplies the "job-mandatory-attribute" operation attribute with the 'digital-signature' keyword value, then the Printer MUST reject the job if the "digital-signature" attribute is supplied with a value that isn't supported by the Printer (or the Printer doesn't support the "digital-signature" attribute at all). The "digital-signature-default" Printer Description attribute MUST be supported if any are. For the IPP Document object spec, the "digital-signature" operation/Document Description attributes and the corresponding "digital-signature-default" and "digital-signature-supported" Printer Description attributes will be OPTIONAL for a Printer to support. If the Printer supports the "digital-signature" operation attribute, it MUST support the "document-format-default" and "document-format-supported" Printer Description attributes as well (and vice versa). For IPPFAX, the Receiver MUST support these attributes. If the Printer supports the "digital-signature" operation attribute and also supports the Document object (not IPPFAX), the Printer MUST support the "digital-signature" Document Description attribute. The implementation MUST support 'none' value if this attribute is supported.
Editor's ISSUE: OK to prefix this attribute with "document-" to make it "document-digital-signature"?

5. Participants

Name	Company	Email Address
Berkema, Alan	HP	alan_berkema@hp.com
Bradshaw, Elliott	Oak	ElliottBradshaw@oaktech.com
Carney, Dennis	IBM	
Farrell, Lee	Canon	Lee.Farrell@cda.canon.com
Hall, David	HP	dhall@hp.com
Hastings, Tom	Xerox	thastings@cp10.es.xerox.com
Kennedy, Kelli		
Lewis, Harry	IBM	harryl@us.ibm.com
Markle, Cathy	HP	Cathy_markle@hp.com
McDonald, Ira	High North	imcdonald@sharplabs.com
Songer, Gail	Peerless	gsonger@peerless.com
Taylor, Bob	HP	bobt@hp.com
Thrasher, Jerry	Lexmark	thrasher@lexmark.com
Tronson, Ted	Novell	Ttronson@Novell.com
Wagner, Bill	NetSilicon	wwagner@netsilicon.com
Zehler, Peter	Xerox	Pzehler@crt.xerox.com