15 April 2011
Candidate Standard 5108.01-2011

The Printer Working Group

MFD Model and Common Semantics

Status: Approved

Abstract: The original PWG Semantic Model [PWG5105.1] has been extended from just the printing service to all of the typical MFD Services, reflecting the consolidation of hardcopy imaging services in Multifunction Devices (MFDs). The MFD model is described in this MFD Model and Common Semantics specification in conjunction with the PWG Semantic Model XML Schema [MFD_SCHEMA], individual specifications for each of the services, and a specification for the System object, the core Element of an MFD. The MFD Model and Common Semantics specification identifies Services that may be included within an MFD, addresses the relations between these Services and the System, describes the concepts and elements common to the different services, and defines the terminology used in the Service specifications.

This Document is a PWG Working Draft. For a definition of a "PWG Working Draft", see:
ftp://ftp.pwg.org/pub/pwg/general/process/pwg-process30.pdf

This Document is available electronically at:
ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm20-mfdmodel10-20110415-5108.01.pdf

Title: **MFD Model and Common Semantics**

About the IEEE-ISTO

The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and flexible operational forum and support services. The IEEE Industry Standards and Technology Organization member organizations include printer manufacturers, print server developers, operating system providers, network operating systems providers, network connectivity vendors, and print management application developers. The IEEE-ISTO provides a forum not only to develop standards, but also to facilitate activities that support the implementation and acceptance of standards in the marketplace. The organization is affiliated with the IEEE (http://www.ieee.org/) and the IEEE Standards Association (http://standards.ieee.org/).

For additional information regarding the IEEE-ISTO and its industry programs visit:

http://www.ieee-isto.org.

About the Printer Working Group

The Printer Working Group (or PWG) is a Program of the IEEE-ISTO. All references to the PWG in this Document implicitly mean "The Printer Working Group, a Program of the IEEE ISTO." The PWG is chartered to make printers and the applications and operating systems supporting them work together better. In order to meet this objective, the PWG will Document the results of their work as open standards that define print related protocols, interfaces, data models, procedures and conventions. Printer manufacturers and vendors of printer related software would benefit from the interoperability provided by voluntary conformance to these standards.

In general, a PWG standard is a specification that is stable, well understood, and is technically competent, has multiple, independent and interoperable implementations with substantial operational experience, and enjoys significant public support.

Contact information:

The Printer Working Group
c/o The IEEE Industry Standards and Technology Organization
445 Hoes Lane
Piscataway, NJ 08854
USA

MFD Web Page: http://www.pwg.org/mfd

MFD Mailing List: mfd@pwg.org

Instructions for subscribing to the MFD mailing list can be found at the following link:
http://www.pwg.org/mailhelp.html

Members of the PWG and interested parties are encouraged to join the PWG and MFD WG mailing lists in order to participate in discussions, clarifications and review of the MFD Working Group product.

This page intentionally left blank.

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Scope

This specification presents the concepts, semantics and structure of a generalized model of the hardcopy imaging services provided by a Multifunction Device (MFD), a hardcopy device also known as a Multifunction Peripheral (MFP), a Multifunction Printer (MFP) or an All-in-One. This specification is both an overall introduction to the PWG MFD Model and a description of concepts and Elements common to several MFD Services. It is intended to serve as an orientation to the separate PWG specifications defining the MFD Model. The root Element of an MFD, (i.e., System) and the individual MDF Services (e.g., Copy, Print) are more appropriately covered in their own specifications. This MFD Model and Common Semantics specification is technically aligned with a named version of the PWG MFD XML Schema [MFD_SCHEMA].

For purposes of this modeling, the services that may be performed by an MFD are:

- Print
- Scan
- Copy
- FaxIn
- FaxOut
- EmailIn
- EmailOut
- Transform
- Resource

This Document defines:

- The overall MFD model including the terminology and concepts used in the MFD Service models.
- The models of an MFD Service, Job and Document
- “Imaging Service” complex Elements, representing structures appearing in several Services but because of XML Schema restrictions, not instantiated in any Service; the appropriate Services have parallel structures that include some Service-specific Elements, and
- Elements common to several Services, eliminating the need to repeat these definitions in each Service specification.

The specific model of each MFD service, the specific interfaces and operations, and the factors unique to each service are discussed in the individual MFD Service specifications.

1.2 Background

Office imaging functions were once limited to copying, formed letter printing and primitive telephone-line based facsimile, each performed by a different device. Impact printers gave way to high quality image printers with complex interpreters and network communication. Optical copying devices were replaced by digital scanners driving image printers. Facsimile matured to wed the digital scanners and image printers with more complex encoding and transports. From the viewpoints of utility, functionality and efficiency, it made sense to integrate these imaging services in a multifunction device.

Although there was increasing commonality in technology, there were very different cultures supporting the manufacture, marketing and maintenance of the different office imaging functions. Slowly, copy and fax functions started appearing in printers; print functionality was added to copiers; and facsimile machines had copy functionality added. Eventually, manufacturers supplied equipment specifically designed to address multiple imaging functions. Utilization of the networking, the massive storage, and the internet capabilities that were brought to the office and home environments allowed further expansion

of imaging device functionality and has made the Multifunction Device the primary hardcopy imaging equipment in enterprise and SOHO environments.

Despite the commonality of technology and the related functionality, the terminology, method of use, and anticipated user interaction of the office imaging functions has been tied to the different cultures associated with these functions and has been slow to coalesce. The PWG V2 Semantics effort defines an MFD model with consistent semantics for capabilities, configuration, operations and states for each of the MFD services. Recognizing the actual and historic distinctions, each service model is described in a separate specification.

1.3 Terminology

1.3.1 Conformance Terminology

Capitalized terms, such as **MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, RECOMMENDED** and **OPTIONAL**, have special meaning relating to conformance as defined in RFC 2119 [RFC2119].

1.3.2 MFD Services Terminology

New terms and terms used in a specific way for this modeling are described in the text of this Document and in the individual Service specifications. For convenience, the following table lists these terms with a summary definition.

The definitions below contain common definitions for service qualified terms. The term '<service>' in any of the definitions below should be taken to be the name of the specific Service being considered (i.e., 'Copy", 'EmailIn', 'EmailOut', 'FaxIn', 'FaxOut', 'Print', 'Scan', 'Transform' or 'Resource') when the term is used in the individual Service specification.

Multi-word terms are presented in this table with normal spacing between words. However, in some places in this specification, and in the individual Service model specifications, the term may be referring to a specific XML Element in an XML Schema. In that case, the spaces between words are omitted in accord with the XML Element tag. In either case, the definition of the term is the same.

Table 1 MFD Services Terminology

Term	Definition
<service>	An identification of the specific Imaging Service involved, sometimes preceding (and sometimes embedded in) terms dealing with Devices, Clients, Jobs and Job Elements, Documents and Document Elements (e.g., Print Document Ticket, Scan Intent, FaxIn Job.) See definition of Service.
Active Jobs	A Service instance specific queue containing all the Jobs that are waiting to be processed or are currently processing.
Automatic Document Feeder (ADF)	A mechanism for handling Hard Copy Documents for scanning. The mechanism selects a media sheet from its bin and passes it to the image acquisition subsystem of the Scan Device. After the Scan is complete the ADF transports the Hardcopy Document's media sheet to its final destination (e.g., output bin, ADF bin.)

Term	Definition
Capabilities	Used in the context of <service> Capabilities, this term refers to those Elements of a Service that can be used in the processing or description of a Job or Document. The models include Job Processing Capabilities, Job Description Capabilities and Document Processing Capabilities. Generally, a Service must be able to report its Capabilities and Job Tickets may include the desired values of these Capabilities Elements to describe the User's Intent.
Client	The local or remote software entity that interfaces with the Job Originator and interacts with an Imaging Service.
Content Region	The area of a Hardcopy Document or Digital Document which is to be processed by an MFD Service. Content Regions are applicable to Scan, FaxOut, Copy and, to an extent, Print Services. For example, a Scan Content Region is the portion of a Hardcopy Document media sheet side to be scanned and converted into a Digital Document. A Copy Content Region is the portion of a Hardcopy Document media sheet side to be printed. Depending upon the Service and the implementation. There can be multiple Content Regions defined for a given media sheet side.
Cross Feed direction	The direction perpendicular to the movement of the Hard Copy Document or the direction that the print head or scanner light bar moves. For scanners that use a technology other that a light bar, this is the direction along which the image data is acquired most quickly. (Also called Fast Scan direction, X) This direction is sometimes referred to as X Feed direction.
Default Job Ticket, Default <service>Job Ticket	A Job Ticket data object that is bound to an instance of an Imaging Service. The Job Ticket Default <service>Job Ticket values are used by the Imaging Service when the Job Ticket for Job being processed does not specify a different value.
Destination	The end point network address of a storage location for a Digital Document output from a Service.
Destination URI	A URI that specifies the recipient of a Digital Document transmission for FaxOut or EmailOut services. It may also contain the URI for the routing to the recipient.
Device	An abstract object representing a hardware component that implements one or more Imaging Services. The term may be preceded with the name of the specific Imaging Service (represented here by <service>). A Device exposes every Subunit on the associated host system involved in performing the functions of the indicated Imaging Service. For example a Print Device Scan Device.
Digital Document	A Document in digitally encoded form as distinguished from a Hardcopy Document. The data input to or output from a Service containing the electronic representation a Hardcopy Document. The Digital Document may also include metadata relative to the Document. Digital Documents are the primary input to Print, FaxOut, EmailOut and Transform Services and the primary output from Scan, FaxIn, EmailIn and Transform services. The Print, EmailOut and FaxOut Services may also have a secondary Digital Document output. See Hardcopy Document.
Directory Service	A software application or a set of applications that stores and organizes information about a computer network's users and resources, and that allows network administrators to manage user's accesses to the resources.
Discovery Client	A software application that performs service or resource discovery on a computer network.
Document	An object created and managed by an Imaging Service that contains the description, processing, and status information of a data object submitted by a User. A Document object is bound to a single Job.

Term	Definition
Document Data	The digitized data submitted by a Job Originator as the Document or portion of a Document to be processed by an MFD service, or as the resulting data from the scanning of Hardcopy Document(s) in an MFD. The images from the scanned Hardcopy Document(s) are encoded in a specified format and stored at a Destination.
Document Repository	A local or remote data store where Digital Documents are stored by or recovered from an MFD Service
Document Resource	A Resource associated with a Document within a Job of an MFD Service is a Document Resource.
Document Ticket	A data object that contains a User's <service>Intent for Document processing and descriptive properties of a Document in a Job. Any Document processing properties in the Document Ticket for a particular Job will override the values specified in the Job Ticket's Document processing properties. The content of a Document Ticket is configured by a User through a <service> Client.
Element	A term used to convey structure and relationships in XML Document instances. An Element can contain both content and Elements. Complex Elements are composed, at least in part, of other Elements.
End User	A User (Administrator, Job Owner, Operator, member of the Owner's group or other authenticated entity) for whom the Job output is intended.
Executable Resource	Executable code that is installed in an MFD system and executed for performing a task. Executable Resource includes two types of resources: Firmware, and Software. (See Firmware, Software definitions below.) Executable resource is a category of resources that is served by the Resource Service.
Fast Scan direction	Same as Cross Feed direction or X.
Feed direction	The direction along which the physical medium is moved or the direction that the print head or scanner light bar moves relative to the Hard copy Document. For scanners that use a technology other that a light bar this is the direction along which the image data is acquired most slowly. (Also called Slow Scan direction or Y)
Firmware	Persistent computer instructions and data embedded in the MFD that perform the basic functions of that device. Firmware is only replaced during a specialized update process. [IEEE2600] Firmware is a type of resource that can be retrieved and stored by PWG MFD Resource Services.
Font	A complete character set of a single size and style of a particular typeface. Most current computer fonts are based on fully scalable outlines. However, the term "Font" still refers to a single style. Times New Roman regular, italic, bold and bold italic are four fonts, but one typeface. Font is a type of resource that can be retrieved and stored by a MFD Resource Service.
Form	A Document (printed or electronic) with spaces in which to write or enter data. Used in the context of the MFD specifications, the term "Form" refers to an electronic form, which is a type of resource that can be retrieved and stored by PWG MFD Resource Services.
Group Element	A collection of Elements that constitutes a complex Element.
Hardcopy Document	A Document on physical media such as paper, transparency or film that is the input source to Scan, Copy and FaxOut Services and the output from Print, Copy and FaxIn Services.

Term	Definition
ICC Profile	A set of data that characterizes a color input or output device, or a color space, according to standards promulgated by the International Color Consortium (ICC). Profiles describe the color Elements of a particular device or viewing requirement by defining a mapping between the device source or target color space and a profile connection space (PCS) (see definition below). This PCS is either CIELAB or CIEXYZ. Mappings may be specified using tables, to which interpolation is applied, or through a series of parameters for transformations. Every device that captures or displays color can have its own profile. ICC profile is a type of resource that can be retrieved and stored by a PWG MFD Resource Service.
Image	A digital electronic representation of the information captured by a Scan Device. One Image is produced as a result of a Scan Device scanning a Scan Region. One or more images are contained in the Digital Document produced by a Scan Service.
Imaging Client	A Client interacting or seeking to interact with an Imaging Service.
Imaging Device	A hardware entity that supports one or more Imaging Services (as defined below), including the System. A more general alternate term for Multifunction Device, which name suggests that more than one Imaging Service is supported.
Imaging Service	One of the Services considered in the modeling of a Multifunction Device, including *Print, Scan, Copy, FaxIn, FaxOut, EmailIn, EmailOut, Transform and Resource*
Intent	The User's preferences for the processing and description properties of a Job or Document.
Job	A data object, created and managed by a Service, that contains the description, processing, and status information of a Job submitted by a User. The Job can contain one or more Document objects.
Job History	An MFD Service instance specific queue containing all the <service>Jobs that have reached a terminating state. The terminating states are defined as Completed, Aborted and Canceled. The length of this queue is determined by the implementer. The Jobs should remain in the Job History for a time sufficient for interested parties to obtain information on completed Jobs.
Job Originator	The User that submits the initial request to create the Job.
Job Owner (or Owner)	Normally the User who submits a Job is the Job Owner, although under certain circumstances an administrator can reassign ownership. The Job owner has certain administrative privileges with respect to his Jobs.
Job Receipt	An Element of the Service that contains information on the actual values of processing Elements used by the Service for processing a Job. The content of a Job Receipt is populated by the Service when a Job is processed.
Job Resource	A Resource associated with a Job.
Job Template	A Job Ticket data object representing a User's preconfigured Intent that is not bound to a specific Service or Job.
Job Ticket	A data object that contains a User's Job-level Intent for Document processing, Job processing and descriptive Job properties of a Job, sent to an MFD Service. Job Elements apply to the entire Job. Document processing Elements apply to all Documents within the Job unless overridden at the Document level (See Document Ticket). The content of a Job Ticket is configured by a User through a Client.
Local Client	A software application entity that is co-located with the Service and interacts on behalf of a User. May also be referred to as Local <service> client, where <service> is one of the Services supported by the MFD.
Logo	A graphical Element, (ideogram, symbol, emblem, icon, sign) that, together with its logotype (a uniquely set and arranged typeface) form a trademark or commercial brand. Logo is a type of resource that can be retrieved and stored by a PWG MFD Resource Service.

Term	Definition
Multifunction Device (MFD)	A hardware entity that supports one or more Imaging Services, including the System. Generally abbreviated MFD, this also includes Imaging Devices commonly called Multifunction Peripherals or Multifunction Printers (MFPs) and All-In-Ones.
Physical Document Ticket, Physical <service>Document Ticket	A printed, encoded Document Ticket submitted by the Job Originator with the Hardcopy Document to be scanned, which becomes a Document Ticket data object after being scanned and processed. This may be used for Scan, Copy and Fax-Out Services.
Physical Job Ticket, Physical <service>Job Ticket	A printed encoded Job Ticket, submitted by the Job Originator with the Hardcopy Document to be scanned, that becomes a Job Ticket data object after being scanned and processed. This may be used for Scan, Copy and Fax-Out Services.
Profile Connection Space (PCS)	A standard device independent color space defined by the International Color Consortium (ICC) that is used for mapping the color space of one device to the color space of another by describing the relationship of each device's color space to this device independent color space.
Region	A rectangular area of a Digital Document that has been specified by an Administrator or End User as the bounding area for which a digital data representation will be output; or a rectangular area of a Hardcopy Document that has been specified by an Administrator or End User to be generated from a Digital Document
Remote Client	The Remote Client is a Client external to the MFD that interfaces with the End User and interacts with a Service.
Resource	A data object that can be served by the Resource Service when required by an MFD system for performing a task or a Job. There are two categories of Resources required by an MFD: Executable Resource, and Static Resource. (See the definitions of Executable Resource and Static Resource).
Resource Client	The local or remote software entity that interfaces with the Job Originator and interacts with a Resource Service.
Resource Service	A software service that provides the interfaces for storing, retrieving, and maintaining users' Resources.
Retained Job	A Retained Job is one which remains in the Service after it has been completed or canceled. This retention may be incidental or because the Job, is a Saved Job. A Retained Job is the only type of Job that can be referenced in a Resubmit Job operation.
Saved Job	A Completed or Canceled Job with a "JobSaveDispostion" Element value indicating that the Job, including Document Data if any, should not be deleted or aged-out after the Job is completed.
Sequence	A type of XML structure that represents an ordered list of Elements.
Service	An Imaging Service (or MFD Service) that accepts and processes requests to create, monitor and manage Jobs, or to directly support other Imaging Services in an imaging-specific way (i.e., the Resource Service). The Service accepts and processes requests to monitor and control the status of the Service itself and its associated Resources. A Service may be hosted either locally or remotely to the MFD.
Slow Scan direction	Same as Feed Direction or Y
Software	Persistent computer instructions and data placed on the MFD, via download, that are separate from, and not a part of, the base Firmware. Software supports features or applications that are in addition to those provided by the base Firmware. Software is a type of Executable Resource. Software is a type of resource that can be retrieved and stored by a PWG MFD Resource Service.
Static Resource	A non-executable electronic data object that is required by an MFD system for performing a task. Static Resource includes the following types of resource: Font, Form, Images, Logo, and Template.

Term	Definition
Subunit	A physical entity contained within or controlled by an Imaging Device that performs a specific process necessary to support one or more Imaging Services; (e.g., Marker, Console, Covers, Interfaces, Input Channels, Output Channels, etc). Note that, for historical reasons, the term Scanner may refer to either a Subunit or a Scan Device.
System	The object handling interaction that needs to be with the MFD as an entity rather than a specific Service. The System is modeled to include all of the Subunits of the MFD, as well as device identification and overall status.
Template	A data object that contains descriptive information and the pre-configured content of a Job or Document Ticket for a specific MFD service. A Template is not bound to a specific Job or Document. It can be stored or retrieved from a Resource Service, collocated on the MFD or hosted on a remote system. Template is a type of Resource that can be retrieved and stored by a PWG MFD Resource Service.
User	Users include the Administrators, Job Owners, Operators, members of the Job Owner's group and other authenticated entities.
Watermark	A recognizable image or pattern when printed on paper used to identify authenticity. Watermark is a type of resource that can be retrieved and stored by a PWG MFD Resource Service.
X	An axis of the coordinate system. This axis is associated with the Fast Scan direction of the Scan Device. If the light bar or Hard Copy Document moves, they do not move in the X direction. They move in the Y direction. (See also Cross Feed direction, Fast Scan direction)
Y	An axis of the coordinate system. This axis is associated with the slow scan direction of the Scan Device. If the light bar or Hard Copy Document moves, they move in Y direction. (See also Feed direction, Slow Scan direction)

1.3.3 Modeling Representations used in this Document

This specification describes the overall aspects of the MFD model in general. Furthermore, it describes in detail the Elements of the model that are common to multiple Services by presenting the graphic representation of the applicable Elements from the PWG MFD XML Schema [MFD_SCHEMA] and providing the datatype, a brief description and the reference for each of the constituent Elements in a table. The references identify the existing standard document, usually a MIB or IPP specification, where the Element is defined. In instances where an Element is newly defined in this specification, the Element definition is provided.

1.3.3.1 Element, Operation and Attribute Name Representation

In the text of this specification, the names of semantic Elements follow normal English format, with the individual words in a multi-word name initially capitalized (as are all defined terms) and the individual words separated by a space. However because the graphic representations are generated from the PWG MFD XML Schema [MFD_SCHEMA], the Elements in the figures and in the tables providing details of these Elements use the XML nomenclature. In the case of multi-word names, the words are concatenated (no spaces) with the first letter of the each word capitalized. For example, the complex Element ‘Subunit Status’ appears this way when being discussed in the text, but appears as ‘SubunitStatus’ in the schema diagrams and their associated tables.

However, multi-word Operation and Attribute names, which do not normally appear in the schema figures, are presented in the text concatenated (no spaces) with the first letter of the each word capitalized (e.g., CreateJob.)

1.3.3.2 "Any" Element and "Any" Attribute Extensions

The MFD model allows for vendor extensions to the model at various levels. Within the schema figures, an "Any" Element symbol is shown as an Element extension point to allow vendor product differentiation by implementation-specific Element extensions while maintaining interoperability. That is, one or more vendor extension Elements may be added where the schema shows an "Any" Element extension. Within the schema diagrams included in this specification, the Element extension is represented by "Any" in a rectangle, as shown in Figure 1.

Figure 1 Element Extension Symbol

In some cases, Attribute extensions are also allowed as Attributes associated with an Element. In the schema diagrams included in this specification, the Attribute extension is represented by "Any" in a five-sided, tab-like symbol as shown in Figure 2.

Figure 2 Attribute Extension Symbol

1.3.3.3 Representation of Mandatory versus Optional Elements and Number of Instances

The line leading to an Element representation may be marked with the minimum and maximum number of instances of that Element. If there is no instance marking or if the minimum instance is greater than 0, that Element is mandatory to the extent that what precedes it is implemented. For example, Figure 3 shows that Copy Service Status is a mandatory element, but it is mandatory only if the preceding Element (which in this case is Copy Service) is implemented. However, because the schema figure segments in this specification are but small parts extracted from the overall model schema, considering the tags on one figure may give an incorrect impression. Since instances of an Element may occur at different places in the model, it is best to view the entire schema when considering these instance tags.

Figure 3 Conditionally Mandatory Element

A tag of 0.1 as in Figure 4 means that the Element is optional, but if it exists, no more than one may exist at that position in the model. A tag of 0.* as shown of the "Any" Element in Figure 1 means that the Element is optional but any number could exist in that position.

Figure 4 Optional Element

1.3.3.4 Representation of Exclusive Or

In this specification, an "or" condition may be represented by a vertical bar (|) not just in an Augmented Backus–Naur Form (ABNF) notation but also when listing parameters that are mutually exclusive, as in "IsAcceptingJobs|IsAcceptingResources".

1.3.4 Datatypes

The modeling of an MFD is schematically represented in PWG MFD XML Schema [MFD_SCHEMA], using a structured hierarchy of components eventually leading down to leaf Elements of simple XML datatypes. In this specification these components are defined in terms of their constituent Elements, and the leaf Elements are identified as being of one of the datatypes in Table 2 Basic Element Datatypes.

When the datatype is keyword or list of keywords (fixed or extensible), the description will identify the keyword group. A fixed keyword datatype is specified as a WKV (Well Known Values), i.e., an enumeration of the allowed values defined in the relevant MFD specification please refer to the keyword group (i.e., PwgWellKnownValues.xsd, MediaWellKnownValues) in the PWG MFD XML Schema [MFD_SCHEMA] for an explicit list of the applicable keyword values. An extensible keyword datatype is usually specified as a union of the WKV and a keyword extension pattern (e.g., KeywordNsExtensionPattern, MimeExtensionPattern). The WKV defines the values that are explicit in the specification and the keyword extension pattern defines the syntax for vendor extended values. Actual vendor extended values are outside the scope of this specification. There are several instances where extensible keywords are defined simply as keyword this is done when the set of explicit keywords is very large (e.g., the set of all the processing and description Element names) and the vendor extensions themselves must be specified as keywords.

While the abstract datatype “list of Xxx” may be unordered for some Elements, the concrete encoding defined is “sequence of Xxx” in XML Schema [W3C-XSD2] and “1setOf Xxx” in IPP/1.1 [RFC2911], both of which are ordered datatypes. Many PWG Semantic Model and IPP/1.1 Elements in fact *do* depend on ordered values, so implementations should preserve the ordering (e.g., in protocol gateways).

Table 2 Basic Element Datatypes

Abstract Datatype	XML DataType	XML Schema Reference	Description and IPP Reference
any	anyType	[W3C-XSD1] 3.4.7	This datatype is used throughout the model to indicate the location of extension points in the model for the PWG or vendors to add Elements in a way consistent with the model. Those who wish to extend the model may do so only at these points.
boolean	boolean	[W3C-XSD2] 3.3.2	This datatype has two possible values: ‘true’ and ‘false’. See: Section 4.1.11 ‘boolean’ in IPP/1.1 [RFC2911]
base64binary	base64binary	[W3C-XSD2] 3.2.16	This datatype consists of a sequence of four characters representing a 24-bit value using an encoding processes defined in RFC 2045 [RFC2045]. The encoding process represents 24-bit groups of input bits as output strings of four encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating three 8-bit input groups. These 24 bits are then treated as four concatenated 6-bit groups, each of which is translated into a single digit in the base64 alphabet.
Complex	complexType	[W3C-XSD1] 3.4.1	This datatype is used to indicate that the associated Element is a container for multiple Elements.
list of complex	sequence of complexType	[W3C-XSD2] 3.4.1	This datatype is used to indicate the associated Element is an ordered set of containers for multiple Elements. See: Section 4.1.16 ‘1setOf X’ in IPP/1.1 [RFC2911]
dateTime	dateTime	[W3C-XSD2] 3.3.8	This datatype consists of a string of characters that represent a year, month, day, hour, minute, second and timezone. This datatype is derived from the ISO definition [ISO 8601:2000 Second Edition]. See: Section 4.1.14 ‘dateTime’ in IPP/1.1 [RFC2911]
hexBinary	hexBinary	[W3C-XSD2] 3.3.16	This datatype consists of a sequence of binary octets encoded in a protocol binding-specific safe manner. This datatype is used for opaque data instead of text. See: Section 4.1.10 ‘octetString’ in IPP/1.1 [RFC2911]
Int	int	[W3C-XSD2] 3.4.17	This datatype consists of a 32-bit signed integer. See: Section 4.1.12 ‘integer’ in IPP/1.1 [RFC2911]

Abstract Datatype	XML DataType	XML Schema Reference	Description and IPP Reference
list of int	sequence of int	[W3C-XSD2] 3.4.17	This datatype consists of an ordered set of 32-bit signed integers. See: Section 4.1.16 '1setOf X' in IPP/1.1 [RFC2911]
range of int	restriction of int	[W3C-XSD2] 3.4.17	This datatype is a complex type consisting of the Elements "upperBound" and "lowerBound" which are both 32-bit signed integers. See: Section 4.1.13 'rangeOfInteger' in IPP/1.1 [RFC2911]
keyword	restriction of NMTOKEN	[W3C-XSD2] 3.4.4	This datatype consists of a string of US-ASCII characters representing a single enumeration value. The legal characters that may be used in defining a keyword are uppercase and lowercase letters, decimal digits, hyphen(-), dot (.), and underscore(_). Colon (:) may also be used, but ONLY for distringuishing namespace prefixes. Vendor extension keywords must be qualified with a prefix of valid characters, followed by a colon (:), and then the vendor's extended keyword. Note that IPP/1.1 [RFC2911] does not allow the use of uppercase letters. IPP keywords are transformed to XML Schema keywords by deleting hyphens and capitalizing the following letter. See: Section 4.1.3 'keyword' in IPP/.1 [RFC2911]
list of keyword	sequence of restriction of NMTOKEN	[W3C-XSD2] 3.4.4	This datatype consists of an ordered set of keywords. See: Section 4.1.16 '1setOf X' in IPP/1.1 [RFC2911]
string	string	[W3C-XSD2] 3.3.1	This datatype consists of a UTF-8 [RFC3629] encoded human-readable string of Unicode characters as defined in [ISO646] [UNICODE]. See: Section 4.1.1 'text' in IPP/.1 [RFC2911]
list of string	sequence of string	[W3C-XSD2] 3.3.1	This datatype consists of an ordered set of strings. See: Section 4.1.16 '1setOf X' in IPP/1.1 [RFC2911]
anyURI	anyURI	[W3C-XSD2] 3.3.19	This datatype consists of a string containing a URI as defined in [RFC3986]. See: Section 4.1.5 'uri' in IPP/.1 [RFC2911]
list of anyURI	sequence of anyURI	[W3C-XSD2] 3.3.18	This datatype consists of an ordered set of URIs. See: Section 4.1.16 '1setOf X' in IPP/1.1 [RFC2911]
anyURI Scheme	restriction of string	[W3C-XSD2] 3.3.1 and 3.3.18	This datatype consists of a URI scheme name as defined in [RFC3986] and registered in [RFC4395], followed by a colon and double slashes (where applicable). See: Section 4.1.6 'uriScheme' in IPP/.1 [RFC2911]
list of anyURI scheme	sequence of restriction of string	[W3C-XSD2] 3.3.1 and 3.3.18	This datatype consists of an ordered set of URI schemes. See: Section 4.1.16 '1setOf X' in IPP/1.1 [RFC2911]

2 MFD Model Concepts

2.1 Top Level MFD Model

The PWG MFD XML Schema [MFD_SCHEMA] is a concise description of an MFD, identifying all Elements in the model about which information may be communicated and/or to which operations may be addressed. In this model, the top Element of the MFD is the System object. As shown in Figure 5, the System includes four mandatory Complex Elements which are basic to the perceived imaging functionality of the MFD, three optional Elements and an extension point.

- Services: The group of Complex Elements performing the functional imaging services associated with an MFD. The Elements common to two or more Services are identified in this specification. Each Service and the Elements and characteristics unique to that Service are individually described in a separate specification.
- System Configuration: the set of Subunits contained in the MFD, certain of which are used by each Service to implement Service functionality. Because the Subunits are a necessary part of the Service specifications, the Subunits are also addressed in this specification.
- System Description and System Status: the Elements of the System object that are administrator settable (description) and inherent (status); these System specific elements are more appropriately described in a separate System specification.
- Managers, Agents and Devices: The Elements involved in managing the Services and the actual or virtual devices implementing them. Although represented in the PWG MFD XML Schema [MFD_SCHEMA], these Elements are discussed in a separate specification. [PWG5106.2].

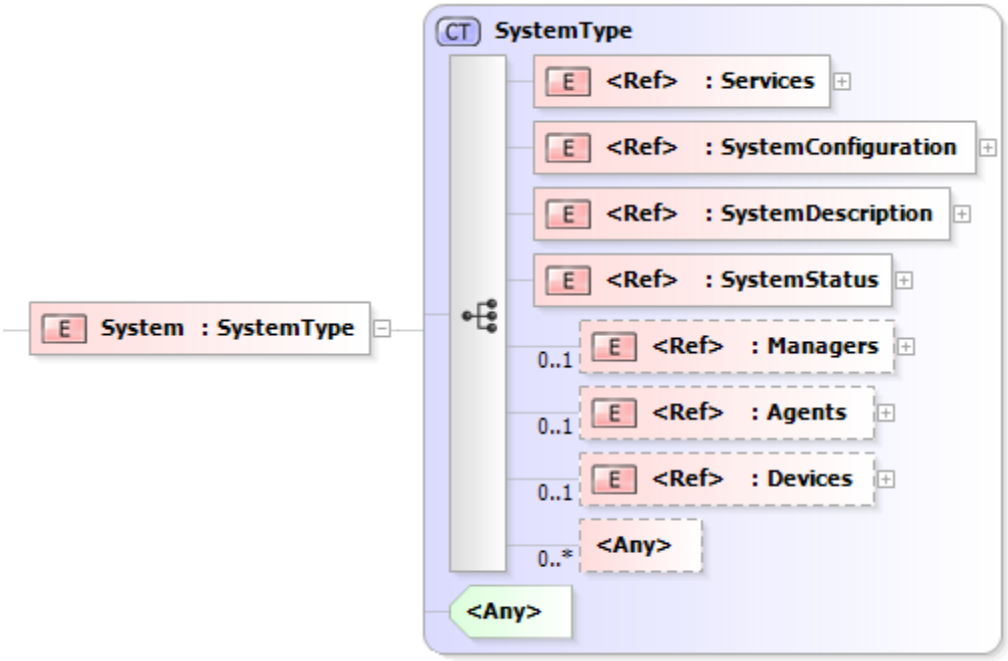

Figure 5 Model of the MFD System Object

2.2 Overview of the MFD Services

The identified MFD Services are modeled to reflect the typical users' perception of imaging services; Print, Scan, Fax etc. These intuitive perceptions can be strengthened by considering the relation between each service and the "outside world", between other services, and between the Service and the MFD itself.

2.2.1 Primary Service Interfaces

The MFD Services and their primary interfaces are represented in Figure 6. All MFD Services other than the Resource Service (and the System Control Service, which is not shown here) process Jobs that deal with Documents. The Documents may be in either Hardcopy Document form, processed by Scanner or Marker Subunits, or electronic (digital) form, communicated through a network, Fax or Email interface Subunit. All MFD Services are integral in themselves, interfacing with an external client, repository or communication facility. (Note that "repository" in this context refers to an external Digital Document source or destination, such as a storage medium.) Although the PWG MFD XML Schema [MFD_SCHEMA] does not address work flow, it does define Service interfaces and operations in a manner which would allow the use of some established workflow methodology.

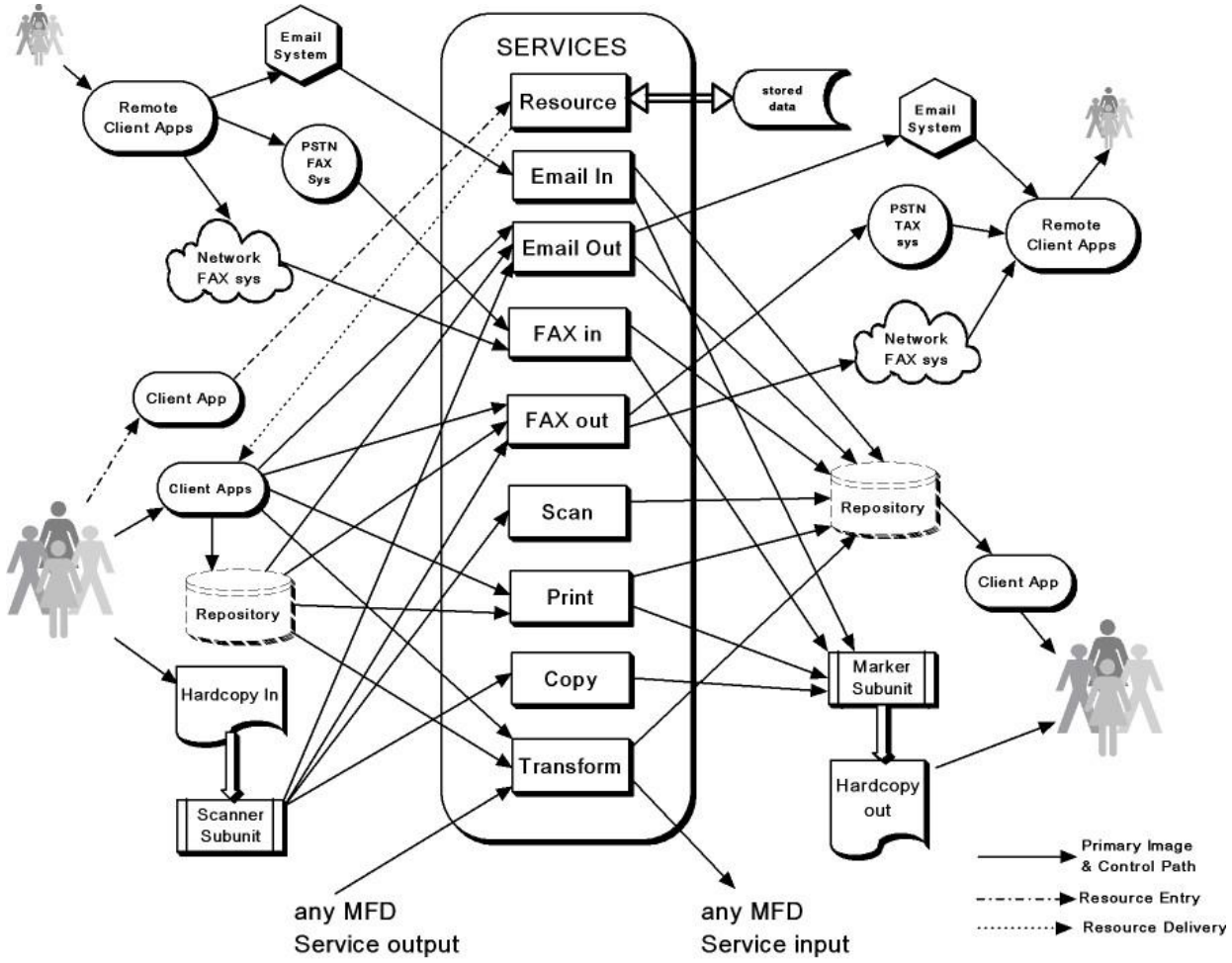

Figure 6 Primary Interfaces with Services

The Resource Service is intended to support the image-processing MFD Services, although it could be used as an independent, but limited, storage/retrieval service. The Resource Service interfaces represented in the figure are those to enter and maintain resources, and those creating a Job Template for submission of a JobCreation request to a Document-processing MFD Service. It would also be possible for the Document-processing MFD Services to include or interface with clients which would directly access resources to be used in a Job, such as fonts or forms, from a Resource Service.

The Transform Service is modeled as an independent primary service, accepting Digital Documents from clients and delivering transformed Digital Documents to clients. It is analogous to the Copy Service that takes in a Hardcopy Document and outputs one or more Hardcopy Documents. The Transform Service may also be used in tandem with one or more other MFD Services in a workflow mode.

All Services actually operate on Digital Documents, using the Scanner Subunit and/or Marker Subunit for the HardcopyDocument/Digital Document conversion. (Note that, for Copy Service, the Digital Document is totally internal and neither it nor its characteristics are accessible outside the service; therefore, for modeling purposes the Digital Documents is not included in the Copy Service.) Users' hardcopy interface with a service is either through a Scanner or Marker Subunit. Users' "softcopy" and control interface is always through a Client, which may be a remote application or may be access via a local Console contained in the MFD. The Repository, which stores Digital Documents before and/or after servicing, may be either contained within the MFD or may be remote (such as a network file server), or may be some combination. The Fax transmission/reception facilities may be fully external (e.g., digital network Fax) or partially internal (e.g., PSTN Fax Modem). For functional modeling of the services, it makes no difference if these Subunits are facilities internal to the physical MFD or not.

2.2.2 Functional Overview of a Multifunction Device

The Semantic Model is very general, covering an MFD hosting anything from one to all of the MFD Services as well as an MFD hosting multiple instances of one or more instances of a given Service. Hosting multiple instances of a Service allows an implementation to expose multiple queues for each service instance, each with its own set of defaults and capabilities.

The top level relationship view of the MFD model is represented in Figure 7, which shows the relationship of Services and Subunits (i.e., System Configuration) within the MFD. The root of the model is the "System", which is more appropriately described in a separate System specification.

System Configuration contains all the Subunits that comprise the MFD. These Subunits are the hardware and software entities used by Services to perform their tasks. While the System has a global view of the Subunits, the individual Services contain a Service specific view. The individual Service specifications describe the Service Configuration that may be applicable to that Service type. The Service Configuration identifies the Subunits of the System that are or may be used by that Service.

Each Service (except the Resource and System Control Services, which do not deal with Jobs) also includes a Service Defaults and a Service Capabilities Element. Service Defaults includes the Default Job Ticket and Default Document Ticket, each containing the default ticket values used by that Service. Service Capabilities contains the Job Ticket Capabilities and Document Ticket Capabilities. These provide the allowed values for the Job Ticket and Document Ticket respectively.

Each Service (except the Resource and System Control Services) may contain zero or more Jobs. Jobs reflect user intent as submitted via Job Tickets and possibly Document Tickets. Jobs waiting to be processed or currently being processed are considered Active Jobs. Jobs that have reached a terminated state (i.e., Completed, Aborted, or Canceled) are under Job History. Whether or not the Job History list is implemented and how long Jobs remain on the Job History list is implementation specific.

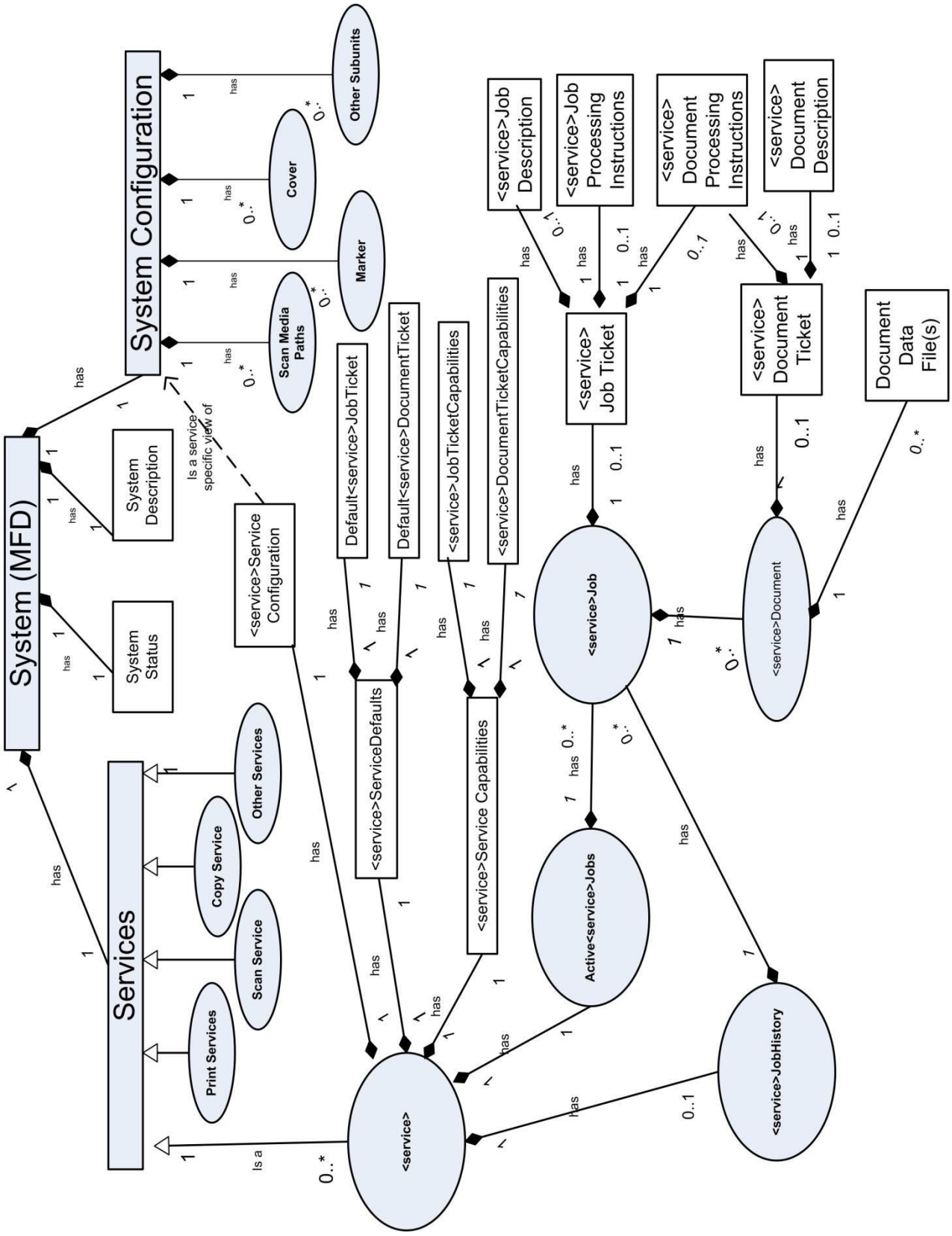

Figure 7 Relationships within a Multifunction Device

2.3 Jobs, Documents, Tickets and Templates

The MFD Semantic Model uses terms in a specific way that may not be familiar. Because these terms and the concepts they represent are integral to the definition of all MFD Services, this section provides conceptual definitions

A Document is text and/or graphic information that initially and/or ultimately is displayed on hardcopy media or on some display device. Documents may be in hardcopy form (Hardcopy Document) or any number of electronic forms (Digital Document). MFD Services handle Documents by:

- Converting Documents between hardcopy and electronic forms (scan and print)
- Duplicating Documents in hardcopy form, usually going through an electronic form (Copy)
- Transmitting Documents electronically (Facsimile)
- Transforming Documents from one electronic form to another (Transform)

A Job is the work Element by which one or more Documents are submitted to a service. A Job includes:

- The Document(s) or reference to the Document(s) which are the objects to be serviced (although, for some Services, there can be a time during Job creation where a Job does not yet have a Document.)
- A Job Ticket, which contains instructions of how the Document(s) should be processed, identification information on the Documents, descriptive information about the Job and optionally one or more Documents, and associated metadata. The Job Ticket Document Processing instructions apply to all Documents within the Job unless overridden at the Document level with a Document Ticket.

Support of multi-Document Jobs is optional. Some services, such as FaxIn, cannot distinguish separate Documents and are effectively limited to one Document per Job. The Service's support for multi-Document Jobs can be determined by examining the Service Capabilities.

Once a Job has been created, its Status may be updated with information about the servicing and the state of the Job. This provides useful for Job management, tracking and billing.

A Ticket is a data object created by a Service and bound to a Job or Document. The ticket contains an end user's Intent for Job and Document processing and the descriptive properties of a Job and optionally one or more Documents.

- A Job Ticket relates to the Job and all Documents in the Job, except those for which a Document Ticket is created.
- A Document Ticket relates to a specific Document and any property specified in a Document Ticket overrides the value of that property that may be specified in the Job Ticket.

A Template, in this context, may be a Job Ticket Template or a Document Ticket Template. It is a partially filled out Ticket not yet bound to a specific Job or Document, but which represents an often-used (or preferred by policy) set of instructions. Templates are completed and/or modified within the service to form specific Job or Document Tickets.

The Job Receipt is produced by a Service. It contains the actual values of processing Elements used by the Service for processing the Job, usually including some information from the Job Ticket.

The Document Receipt is produced by a Service for Documents for which the user has supplied a Document Ticket. The Document Receipt contains the actual values of processing Elements used by the Service for processing the Document, including information from the Job Ticket.

2.4 Content Region and Images

An MFD deals with Documents, converting them from Hardcopy to Digital Document, Digital to Hardcopy Document, Hardcopy to Hardcopy Document or Digital to another form of Digital Document. There is not always a 1:1 relation between a Hardcopy Document and the Digital Document derived from it or from which it is derived. Services that can obtain input from a Scanner Subunit (e.g., Scan, Copy and FaxOut) may allow the user to select or may restrict regions of the original hardcopy to be rendered in the Digital Document. Similarly, Services that provide output to a Marker Subunit may allow selection or may restrict regions of the Digital Document to be rendered in the Hardcopy Document output.

2.4.1 Content Region

Content Region is a complex type referring to a rectangular area of interest in a Document; i.e., an area of the Document that contains desired content. In the MFD modeling, the Scan Region and ImageBox complex Elements are of the Content Region type. The Elements in the Scan Region, which also constitute the constituent Elements of the Content RegionType, are shown in Figure 8.

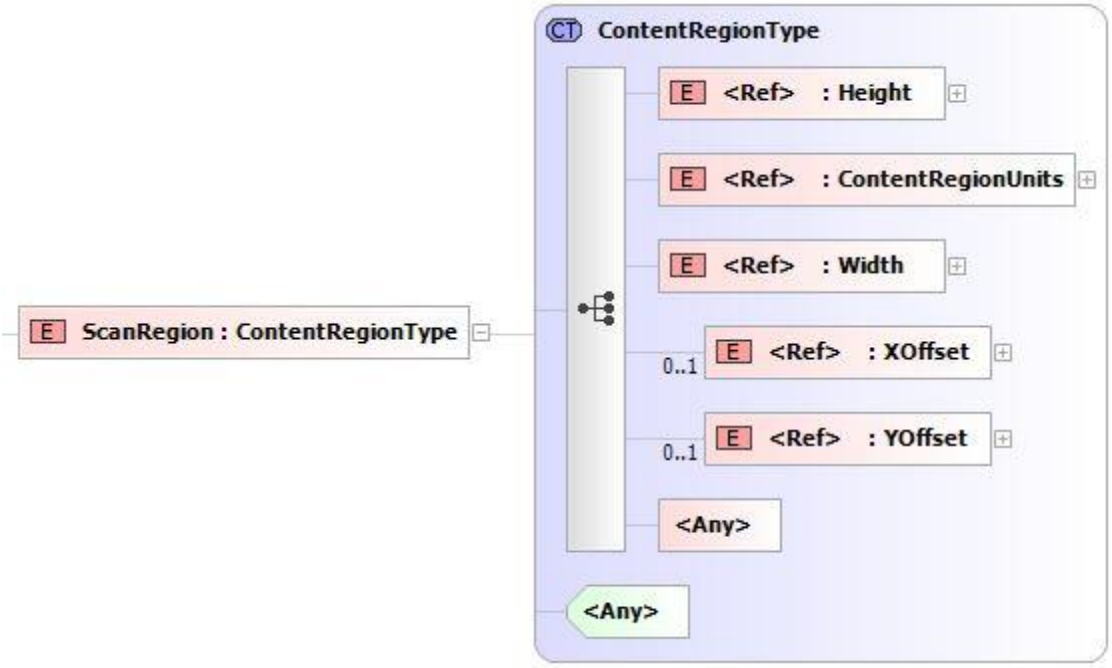

Figure 8 Scan Region Element

It is possible to identify multiple Scan Regions on a media sheet side. Each such region has a one-to-one mapping to a scanned image. The relationships among Hardcopy Document, Scan Region, and Image are represented in Figure 9. A Service using the Scanner Subunit defines a Scan Region and converts the acquired data into an Image. The encoding of the Image at this point is implementation specific. The Service accumulates the set of images that are to be extracted from the input Hardcopy Document and encodes them in the format reflecting appropriate to the Service and the User Intent to produce the Digital Document Depending upon the Service, this Digital Document may be sent to a repository (Scan), to a printer (Copy), or transmitted as a Facsimile (FaxOut).

Content Region also applies to a Service using the Marker Subunit, such as Print. In most cases, the selection of Content Region in these cases is done outside of the MFD. However, in certain production printing environments, there are service Elements to define a print content region independent of the maximum printable area defined as part of the Marker Subunit status. These Elements are included in the Print Service model, but being limited to a special type of Print Service, are not described here. See Production Printing Attributes Set 1[PWG 5100.3], Para. 2.2 Finished Page Image Cell.

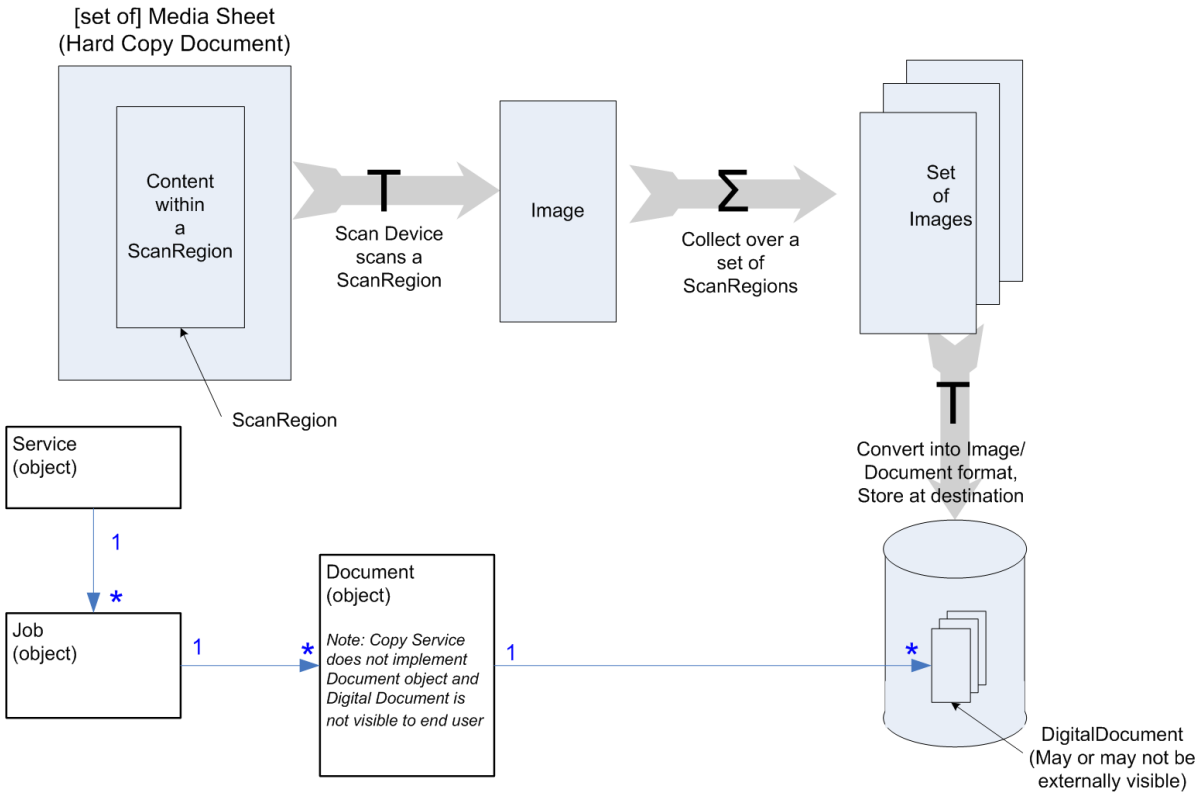

Figure 9 Hardcopy Document, Region, Image Relationship

2.4.2 Job/Document Object and Digital Document Cardinality

There can be multiple Documents in a Job. Digital Documents are output as Files. Therefore, there are various ways in which the Services that accept multi-Document Jobs (i.e., Scan and Transform) can provide their Digital Document output. These are diagramed in Figure 10.

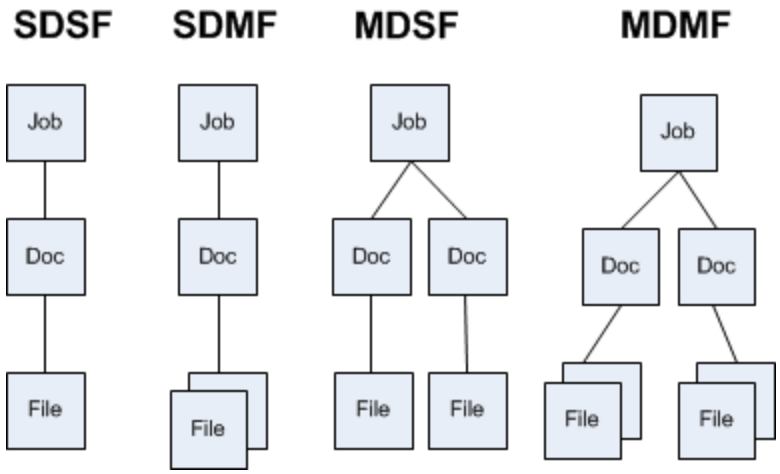

Figure 10 Document Cardinality

SDSF (Single Document Single File Job): One Document object that contains a reference to one single file. There is one file URI for the single Document location. The Destination is the Document URI; i.e., the single file URI.

SDMF (Single Document Multiple File Job): One Document object that contains references to multiple files. Each image in the Document data is stored in a separate file. There is one file URI per image file which constitutes the Document. The Destination is the directory URI of the files for the Document.

MDSF (Multiple Document Single File Job): Job object contains multiple Document objects. Each Document can have a different set of processing parameters. Each file contains a specified number of images for the Document. Each such set of images is stored in a separate file. Each Document object contains one file URI for each image file location. Each Destination is a Document URI; i.e., the single file URI.

MDMF (Multiple Document Multiple File Job): Job object contains multiple Document objects. Each Document can have different set of processing parameters from another. Each image in the Document data is stored in a separate file. The Document object contains a file URI for each image file location. Each Destination is the directory URI for the Document.

A Scan or Transform Job produces 1 to N Document objects. The Job has a unique ID within a unique Service. The Documents within the Job are numbered 1 to N. The Destination of a multi-Document Job is a directory for multi-Document files. After the Document file is stored, the Document file URI will be updated in the Document object as the Document location.

In addition to URI, the MIME type of a stored Document MUST be specified by the end user to clearly indicate the content type to be used for transporting the stored Document.

The capability of a Service to generate the various Document/File cardinalities is declared in the Document Output Mode in the Job Processing Capabilities Element. The value of the Document Output Mode which is set in the Job Ticket allows an End User to request one of the supported values.

2.5 Coordinate Systems

All primary MFD services deal with Documents that, at some point in their existence, are visually presented for human consumption. Ideally, MFD Services would use a consistent coordinate system mapping between a hardcopy (or monitor displayed) Document and its Digital Document form. However, different systems have evolved in different environments and a realistic coordinate system approach must accommodate the existing practices. This section discusses the relationships between the coordinate systems used for scanning, printing, and for various Digital Document formats.

2.5.1 Coordinate Nomenclature

In a comprehensive MFD model, the intent is that the definition of a semantic Element be consistent across the modeling of the constituent Services. Because this also applies to the coordinate system semantics, there may appear to be some inconsistency in the names associated with coordinate related semantics.

There are scan Subunit specific semantics (e.g., Fast Scan Direction), print Subunit semantics, media related semantics (e.g., width), as well as general image processing semantics (e.g., X). Each semantic system has terms which are related to terms in the other systems. The terms Cross Feed, XFeed, Fast Scan, X, and Width are semantically aligned. For Scan and Print Subunits, these terms refer to the axis which is perpendicular to the movement of the Hard Copy Document, the Scan Subunit's light bar or the Print Subunit's print head or laser sweep path; it is also the axis along which the image data is read or written most quickly. The terms Feed, SlowScan, Y, and Height are semantically aligned. They refer to the axis which is parallel to the movement of the Hard Copy Document, the Scan Subunit's light bar or the Print Subunit's print head or laser sweep path; it is also the axis along which the image data is read or written most slowly.

2.5.2 Content Coordinate Systems

Below are descriptions of the coordinate systems and units of measures for the Scanner and Marker Subunits and various Services. The Digital Document output is discussed as it relates to the semantic of the layout affecting Document Processing Elements.

2.5.2.1 Scan and Marker Subunit Media Content Coordinates

Figure 11 identifies the terms used to describe the placement and relative motion of media on a hardcopy scanner or printer. The outermost rectangle represents the surface upon which the media moves in the case of a scanner ADF or typical printer, or rests in the case of a flatbed scanner. The dimensions of this are given in terms of a MediaPathMax Element for printers and a Scan Media Path for scanners. In the case of a flatbed scanner, the platen is considered an instance of the Scan Media Path. These dimensions determine the maximum sized media that may be accommodated.

The media content (represented by the innermost rectangle in Figure 11) is the information that is being impressed upon the media or read from the media. The media itself is represented by the middle rectangle. The leading edge of the media is referred to as the North (N) edge, with the side of the media containing the content area facing the observer. The other edges of the media are defined by the normal compass layout of East (E), West (W), and South (S), with the compass facing the observer.

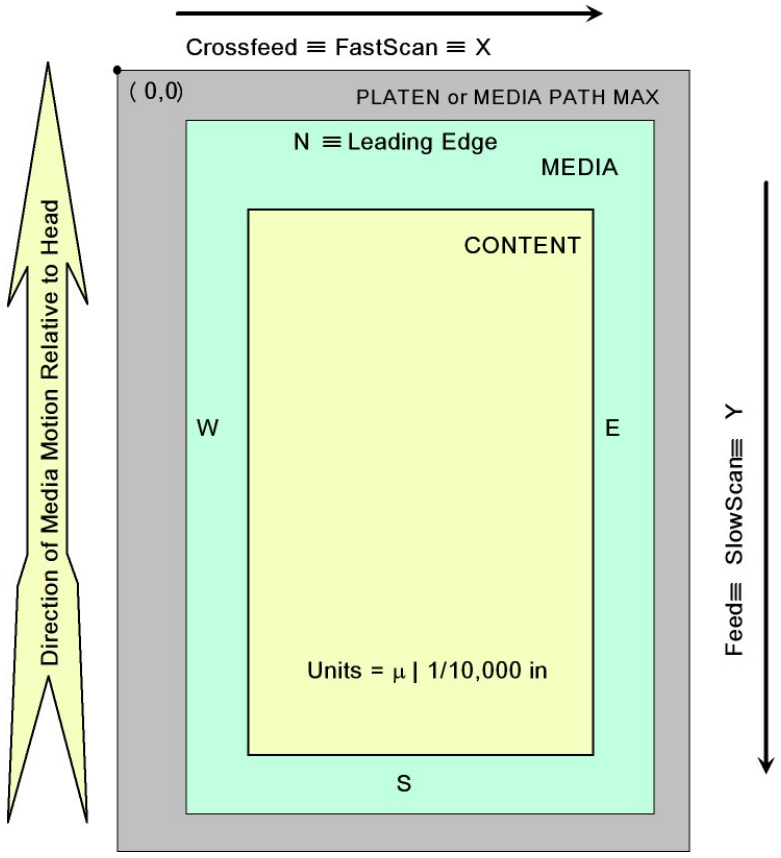

Figure 11 Scanner or Marker Coordinates

In a Scanner ADF or a typical printer, the media is moved in the direction indicated by the large arrow in Figure 11; in a flatbed scanner, the media may be stationary and the scan head moved in the opposite direction. In either case, the relative direction of motion of the media with respect to the head is the same as is the direction of feed or slow-scan (as indicated by the arrow at the right of the diagram.) The content area is scanned in two orthogonal directions, variously defined as Cross Feed, Fast Scan or X and Feed, SlowScan or Y. The origin is at the leading edge of the Cross Feed and proceeds down the media.

2.5.2.2 Service Coordinates

Figure 12 shows the coordinate system that is used for a Service using a Scanner Subunit. The Content Region is facing the observer. Note that a Content Region for a Scan Service is termed a Scan Region (para. 2.4.) The service coordinate system and the defining parameters are the same in both cases. The Scan Region associated with a scan operation is defined in terms of the offset from the junction of the Fast Scan and SlowScan origins and the height and width dimensions of the region. The coordinate system for the Device and Service are linked together; that is, the origin of the Device's coordinate system is the same as the origin of the Service's coordinate system. The Scan Region may be defined by the user, or it may be default to the media size or the maximum allowable scan area. It is possible to have multiple overlapping Scan Regions.

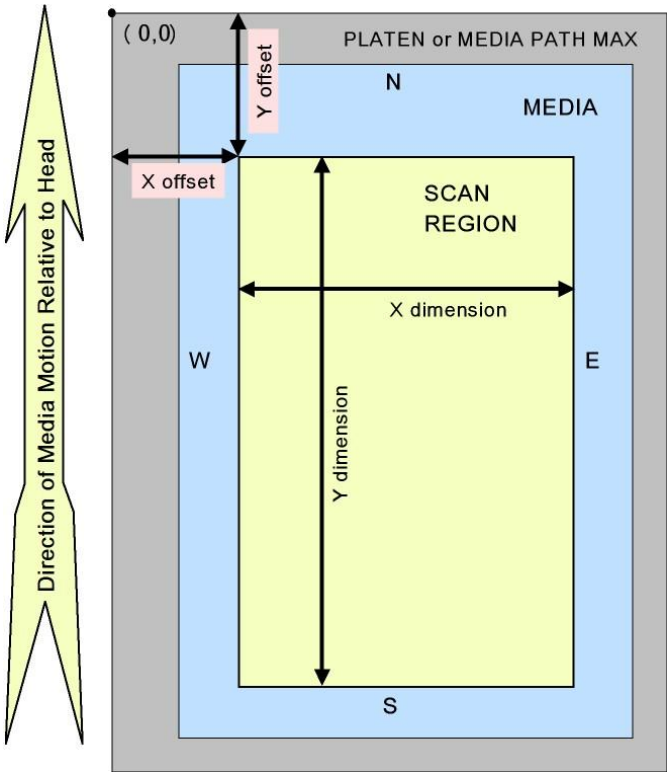

Figure 12 Service Scan Coordinates

2.5.2.3 Document Format Coordinates

When the output of a Service such as Scan or Transform is converted into certain Document Formats and stored (e.g., PDF, XPS), under some circumstances the entire image may not be stored. This is because these Document formats contain various constructs that bound the area where the image can be placed.

As represented in Figure 13, the MFD Model includes the Image Box and the Media Box. The Image Box is where the image will be placed and corresponds to the Hardcopy Document “content area” within the Digital Document. The Media Box delineates the area where the image can appear in the output Document and corresponds to the Hardcopy Document Media Size that may or may not be assumed in the Digital Document. If the X Offset and Y Offset are absent or set to 0, the origin of the coordinate systems for the Image Box and Media Box are aligned.

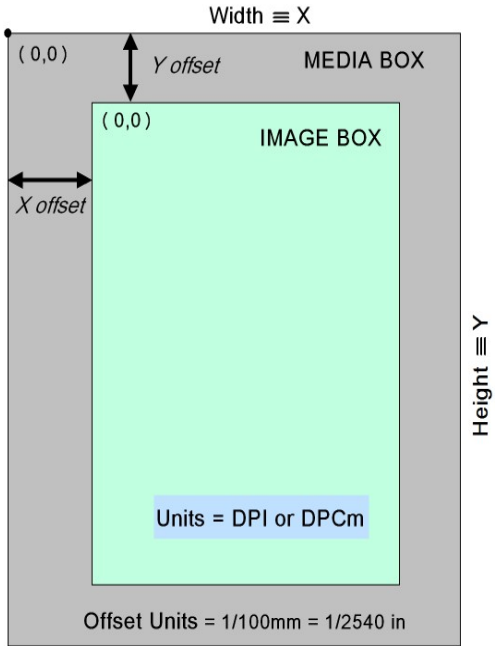

Figure 13 Document Format coordinates

These bounding boxes correspond to two PostScript bounding boxes.

- **ImagingBBox:** A bounding box defining an outer boundary for each page image. This corresponds to the MFD Model Image Box.

- **PageSize:** A rectangular area corresponding to the overall dimensions of the physical medium that was assumed in page description. This corresponds to the MFD Model Media Box.

The Portable Document Format (PDF) defines five boxes described by various bounding conditions:

- **ArtBox:** A rectangle, expressed in default user space units, defining the extent of the page's meaningful content (including potential white space) as intended by the page's creator. This corresponds to the MFD Model Image Box

- **MediaBox:** A rectangle, expressed in default user space units, defining the boundaries of the physical medium on which the page is intended to be displayed or printed. This corresponds to the MFD Model Media Box.

PDF also defines some additional bounding boxes than are not included in the MFD Model.

- **BleedBox:** A rectangle, expressed in default user space units, defining the region to which the contents of the page should be clipped when output in a production environment

- **CropBox:** A rectangle, expressed in default user space units, defining the visible region of default user space. When the page is displayed or printed, its contents are to be clipped (cropped) to this rectangle and then imposed on the output medium in some implementation-defined manner.

- **TrimBox:** A rectangle, expressed in default user space units, defining the intended dimensions of the finished page after trimming.

2.6 Job and Job Ticket Life Cycle

The relationships among the Elements associated with a service are shown in Figure 14. This discussion is applicable to Scan, Print, Copy, FaxOut and Transform MFD services, although aspects of these relationships specific to each service will be discussed in each specific Service specification. The Document and Document Ticket lifecycle is similar to the Job and Job Ticket lifecycle. The main differences are that:

- A Document Ticket contains only Document Description and Processing Elements,
- A Document Ticket is bound to a Document and,
- The Document creation operations are Send Document and Add Document.

The Job Ticket is a data object that contains Job descriptions and Job and Document processing instructions. The Job Ticket life cycle is bound to the lifecycle of its associated Job. The Job is created when a Client initiates a Create Job operation. The Create Job can come from a Remote User interacting with an Application or by a Walkup User interacting with a User Interface and pressing the “START” button. In either case the user is able to communicate his Intent through the <service> protocol. The general sets of operations available to the User and to the Administrator are described in Section 0.

Prior to the Create Job submission, a Job Template can be selected or created. If available, a preconfigured Job Template can be retrieved from a data store outside the scope of this specification. For example the Resource Service can be used to retrieve a Job Template from such a repository. Alternatively a Job Template can be created by instantiating the associated XML Schema or retrieving the Default Job Ticket from the Service.

A Job Template is a Job Ticket that is not bound to any Job or Service. It is used as a convenience or to standardize the submission of future Jobs. A Job Template can be created in a number of ways. Some examples are:

- A walk-up end user uses the Local application and User Interface to create a Job Template by configuring a set of preferred values for Job and Document Processing and Job Description.
- An end user uses a remote application (in a computer) to create a Job Template by configuring a set of preferred values for Job and Document Processing and Job Description.
- A walk-up end user or Remote user creates a Job Template by retrieving the Services Default Ticket and optionally modifying the values.
- A walk-up end user or Remote user retrieves a Job Template from a Resource Service and optionally modifies the values.

The Job Template can then be modified to meet the user’s Intent. The Service also makes available the Service Capabilities that list all the allowed values for a Job’s Job Ticket for that instance of the Service. Once the Job Template fully embodies the User’s Intent, it can be bound to a Create Job operation parameter to send to the Service.

The Service creates a Job based on the parameters in the Create Job operation. The Job’s Job Ticket is created based on the user supplied parameters. The Job Ticket remains bound to the Job until the Job is eventually deleted.

The Services’ Service Job Ticket Capabilities can be retrieved to provide the set of allowed values when modifying the Job Template

If an end user does not select a Job Template, then the processing instructions are the Default Job Ticket values. The Default Job Ticket can be administratively modified to control default behaviors for the associated Service.

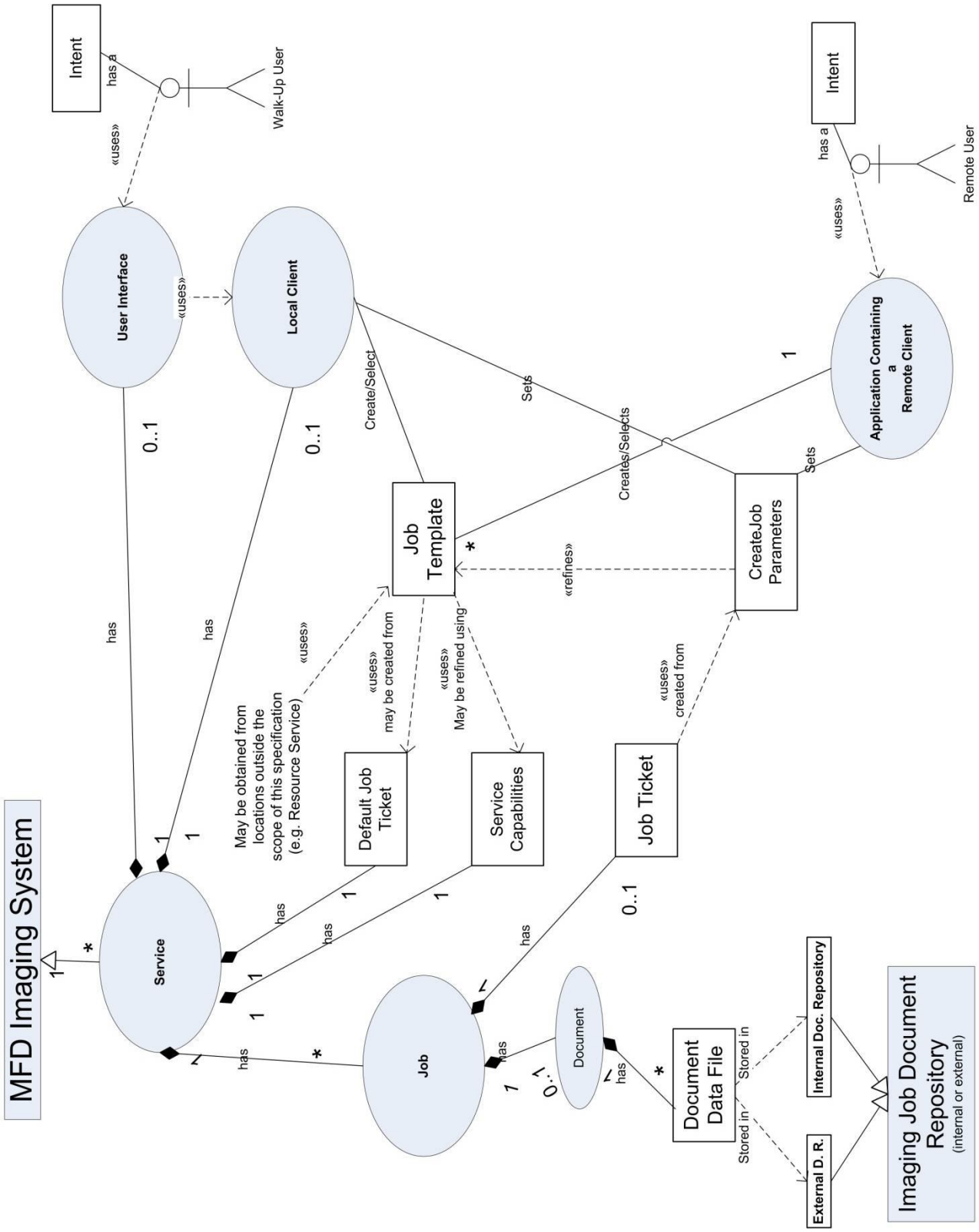

Figure 14 UML Diagram Showing Relationships Relative to Job Ticket

When a Job Ticket is created, it is bound to the end user's Job by the Service. The Service manages and processes the Job according to the instructions supplied in the Job Ticket. If a processing instruction in the Job Ticket cannot be supported, the Service either returns an error or substitutes the instruction with another that best matches the user's intent. The actual values used for processing the Job are captured in the Job Receipt of the Job in the Job History queue, which can be queried later.

After a Job is completed, it is moved to the Job History queue. A Job is deleted from the Service when it is deleted from the Job History according to a set of site-specific Job History retention rules.

The Job Ticket lifecycle ends as soon as the Job Ticket is deleted along with the Job.

3 System Configuration Subunits

The System Configuration represents device semantics in the MFD Model. The included Elements are semantically aligned with objects in the Printer MIB [RFC3805] and the Elements in the DMTF CIM Printing classes defined in [CIM-SCHEMA].

The System Configuration includes all of the Subunits used (or that could be used) by any of the Services in the MFD. For purposes of the model, Subunits are considered in terms of how they relate to Service functionality and Job processing, including configuration and statistics.

Figure 15 shows and Table 3 describes the Subunit Types that may be part of System Configuration and each <service>Configuration. These types are derived primarily from the groups used in the Printer MIB [RFC3805] or the HR MIB [RFC2790] as referenced in the Printer MIB. The System may have 0 to 1 instances of each Subunit type. If a Subunit type exists, there may be one or more instances of the Subunit (although, in the case of some Subunits, such as Cover, that may not be applicable.) No Service uses all possible Subunits; rather, each Service instance has a <service>Configuration which identifies the Subunits that may be used by an instance of that Service.

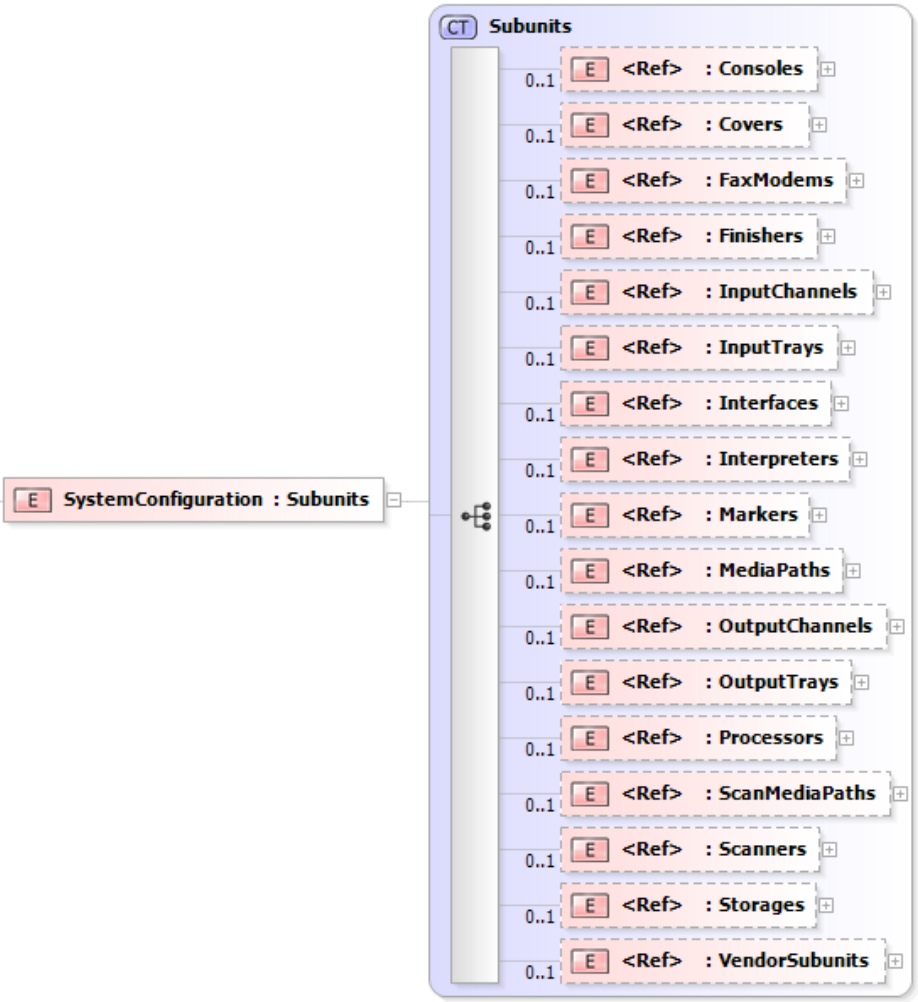

Figure 15 Elements in System Configuration

Each instance of a Subunit has Status and Description complex Elements. The values of Status Elements are determined by the nature or state of the equipment and cannot be directly set by the user or administrator. Description Elements can be set by the users or administrator in Subunit or system configuration.

Table 3 MFD Subunits

Subunits	Reference	Description
Consoles	prtConsole [RFC3805]	User Interface Console(s) used to display and modify the state of the Device, Service, and Job. It may also use for setting configuration, generating and modifying Job Tickets and getting Service history and statistical information
Covers	prtCover[RFC3805]	covers, doors, housings and interlocks.
FaxModems	[RFC1696]	PSTN modems used to transmit and receive Digital Documents
Finishers	prtOutput and all of RFC3806 [RFC3806]	performs some operations on the media other than marking
InputChannels	prtChannel [RFC3805]	source of Job (or Resource) control information and, for some services, the input of Digital Documents
InputTrays	prtInput [RFC3805]	mechanisms that feed media to be marked on into the Media Path
Interfaces	RFC2863 [RFC2863]	communication ports and associated protocol stacks into and out of the Server. The Input and Output Channels utilize the Interfaces.
Interpreters	prtInterpreter [RFC3805]	conversion of Digital Documents into images that are to be marked on the media, of scanned images into Digital Documents of the desired format, and Digital Documents from one format to another.
Markers	prtMarker [RFC3805]	produces marks on media. Includes both the mechanism and the associated supplies
MediaPaths	prtMediaPath [RFC3805]	the mechanisms that move the media which is to marked and connects all marking related Subunits: Input Trays, Output Trays, Markers and Finishers.
Output Channels	prtChannel [RFC3805]	destination pathways for Service Request responses and, for MFD Services with a Digital Document output; the destination pathways for these Documents.
OutputTrays	prtOutput [RFC3805]	general purpose trays capable of receiving Hard Copy Documents that have been printed or scanned. This Element may not be applicable to Scanner Subunits equipped with an ADF that re-circulates the media if the ADF is the only destination of the Hard Copy Documents handled by the ADF.
Processors	RFC2790[RFC2790] hrProcessorTable	computing and logical Elements of device, including associated operational memory, that typically execute the device software and/or firmware
ScanMediaPaths	[PWG5108.2]	encompasses the mechanisms that move the media to be scanned This can include an Automatic Document Feeder for Scanner Subunits so equipped.
Scanners	[PWG5108.2] -	mechanism that acquires the image of the selected scan of a Hardcopy Document
Storages	prtStorage and RFC2790[RFC2790] hrStorage	mechanism that stores and allows retrival of Digital Documents and/or resources
Vendor Subunits		Implementation-specific Elements for which status and description information should be available

Many Status Elements and Description Elements are common to all Subunits. Other Elements reflect the specific nature of the Subunits type. In the subsequent paragraphs of this section, the Elements constituting each Subunit are outlined in Schema graphic segments and the Subunit-specific Elements are described in tables. Note that, at various levels, an “Any” Element is provided as an Element extension point to allow vendor product differentiation by implementation-specific Element extensions while maintaining interoperability. This Element of type “Any” appears through the models to provide vendor extensibility.

3.1 Common Subunit Elements

The Status and Description Elements common to most Subunits are grouped into the complex types Subunit Status and Subunit Description, respectively. Following the MFD modeling convention, Subunit Status contains Elements which define the basic characteristics, state or condition of the Subunit; the values of Status Elements are not subject to administrator modification or reset. Subunit Description contains Elements which are intended to be set by the administrator. The descriptions of these common Subunit Elements are presented once in this section rather than being repeated for each Subunit. The individual descriptions of Subunits that include these Subunit Status and Subunit Description complex Elements refer back to this section.

There will typically be Power Management Elements, derived from the PWG Power Management Model [PWG 5106.4] associated with and operating on the System (i.e., on the MFD as a unit). The description of these complex Elements is appropriate to the System specification. However, within the MFD model, the Subunits correlate most closely to the physical components of the MFD. Because the various Services in an MFD which employ the Subunits may be used at different times and under different circumstances, it may be desirable to have independent power control of the various Subunits. Therefore, the Subunit Status and Subunit Description complex Elements include Power Status and Management Elements. Because of this inclusion, the common Subunit Element groups are not constituent Elements of Subunits for which Power Management Elements are considered inapplicable, such as Covers and software-based Subunits including Interpreters, Input Channels and Output Channels.

3.1.1 Common Subunit Status Elements

The Subunit Status complex type is represented in Figure 16 and described in Table 4. Subsequent subsections describe the constituent power-related complex types. Subunit Status Elements are inherently read-only.

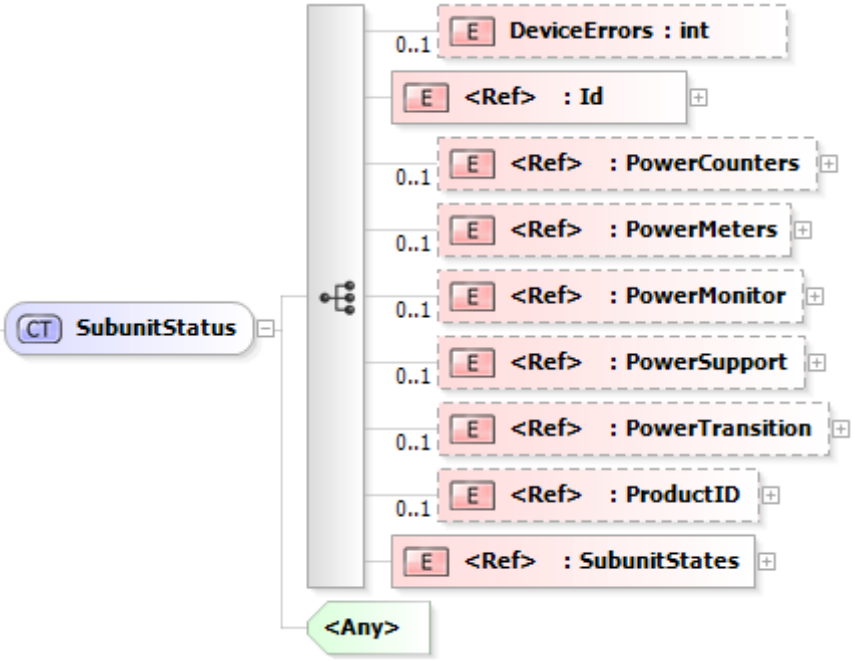

Figure 16 SubunitStatus Elements

Table 4 Description of SubunitStatus Elements

Element	DataType	Description or *Keyword Group*	Reference
DeviceErrors	int	count of detected errors may or may not be power persistent	hrDeviceErrors **[RFC2790]**
ID	int	unique value for each instance of the Subunit in the MFD	hrDeviceIndex **[RFC2790]**
PowerCounters	complex	See 3.1.1.1	
PowerMeters	complex	See 3.1.1.2	
PowerMonitor	complex	See 3.1.1.3	
PowerSupport	complex	See 3.1.1.4	
PowerTransition	complex	See 3.1.1.5	
ProductID	string	manufacturer's ID for the Subunit	hrDeviceID **[RFC2790]**
SubunitStates	keyword	current operational state of the Subunit *SubunitStateWKV*	hrDeviceStatus **[RFC2790]**

3.1.1.1 Power Counters

Power Counters track major power state transitions. Hibernate, On, Standby and Suspend are stable power states, and are defined in an Imaging Device context in the PWG Power Management Model [PWG 5106.4]. The Power Counter Element is represented in Figure 17 and its Elements identified in Table 5. These counters are specified as persistent across power cycles and hardware reconfigurations.

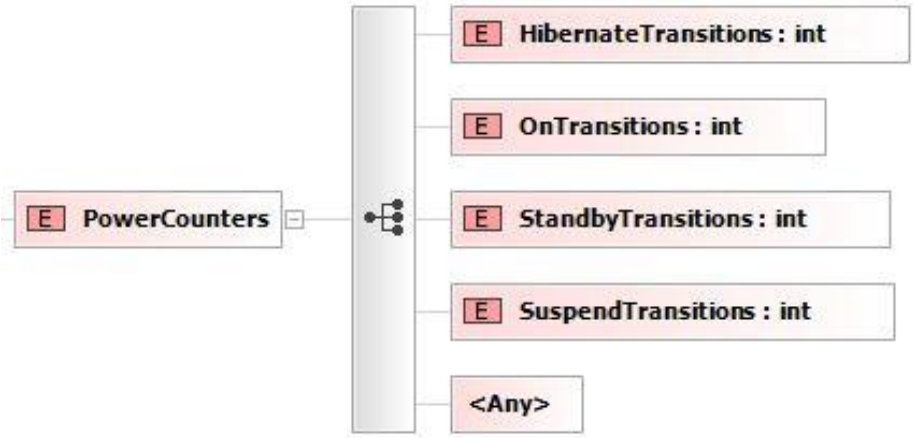

Figure 17 Power Counters Schema

Table 5- Power Counter Elements

Element	DataType	Description	Reference (PWG Power Management Model [PWG 5106.4]
Power Counters	complex		Para 5.4
HibernateTransitions	int		Para 5.4.1
OnTransitions	int		Para 5.4.2
StandbyTransitions	int		Para 5.4.3
SuspendTransitions	int		Para 5.4.4
Any	Various	Extensions to Power Counters	

3.1.1.2 Power Meters

Power Meters includes a group of meter Elements concerned with keeping track of the power consumption of the Subunit. This complex Element is represented in Figure 18, with the component Elements identified in Table 6. The Power Meters Are Actual Element is a Boolean allowing the use of either measured or anticipated power consumption values. Power Peak Watts tracks the maximum energy drawn by the Subunit since the last reboot and is inherently not power persistent. The Power Current Month KWH Element accumulates power consumption values (or estimates) in real time. It may be expected to be continually increasing and to be power persistent during the month. At the end of each month the value is transferred to the Power Previous Month KWH Element value. Power Lifetime KWH is, of course, persistent over power and reconfiguration cycles.

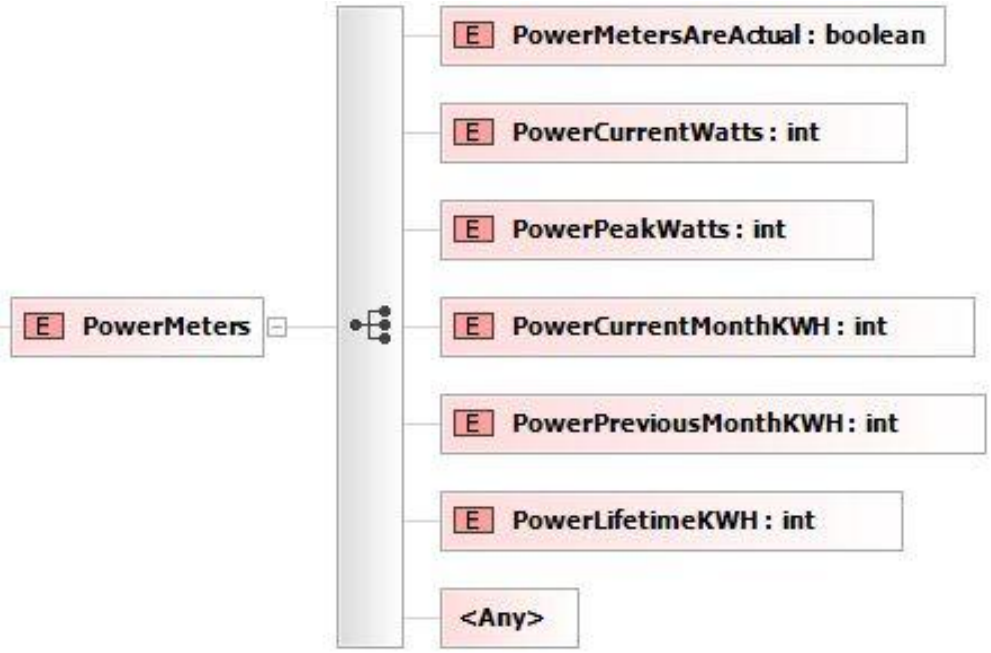

Figure 18 Power Meters Schema

Table 6- Power Meter Elements

Element	DataType	Description or *Keyword Group*	Reference (PWG Power Management Model [PWG 5106.4]
PowerMeters	Complex	the current power meters for each System or Subunit	
PowerMetersAreActual	boolean	power meter values are actual measurement(true) or estimation (false)	Para 5.5 [PWG 5106.4]
PowerCurrentWatts	int	current power consumption in watts	Para 5.5.1 [PWG 5106.4]
PowerPeakWatts	int	peak power consumption in watts	Para 5.5.2 [PWG 5106.4]
PowerCurrentMonthKWH	int	current month's power consumption in kilowatt hours	Para 5.5.3 [PWG 5106.4]
PowerPreviousMonthKWH	int	previous month's power consumption in kilowatt hours	Para 5.5.4 [PWG 5106.4]
PowerLifetimeKWH	int	lifetime power consumption in kilowatt hours	Para 5.5.5 [PWG 5106.4]
Any	Various	Extensions to PowerMeters	

3.1.1.3 Power Monitor

The Power Monitor Elements for a Subunit identifies the current power state of the Subunit. This complex Element consists of just a Power State Element and a Power State Message Element. The representation is Figure 19 and the Elements are identified in Table 7.

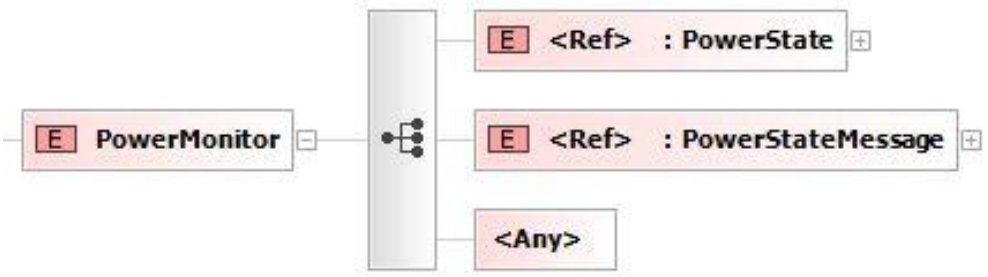

Figure 19 Power Monitor Schema

Table 7 Power Monitor Elements

Element	DataType	Description or *Keyword Group*	Reference (PWG Power Management Model) [PWG 5106.4]
Power Monitor	complex		
PowerState	keyword	*PowerStateWKV*	Para 5.2.1 [PWG 5106.4]
PowerStateMessage	string	human-readable that describes, explains, or qualifies the current power state	Para 5.2.2 [PWG 5106.4]
Any	various	Extensions for Power Monitor	

3.1.1.4 Power Support

The Power Support complex Element lists the supported stable power states for the Subunit along with the functional and power consumption characteristics of the Subunit in each state. The complex Element is represented in Figure 20 and the constituent Elements identified in Table 8.

The Can Use Interfaces Element specifies the Interfaces, listed in a string, that are not inherently disabled when the Subunit is in the identified state, not necessarily that these interfaces can be used. That is, this Element does not report the disabled or down conditions of the Interfaces. Although identified as a string datatype, the value must be in the format “<ID1>,…,<IDn>” (e.g., 1,3,4) and must contain a comma-delimited list of Interface ID values (XML Schema) or corresponding if Index values (SNMP MIB). When this Element is instantiated on a Subunit, it must have the same value as corresponding Element in the System object.

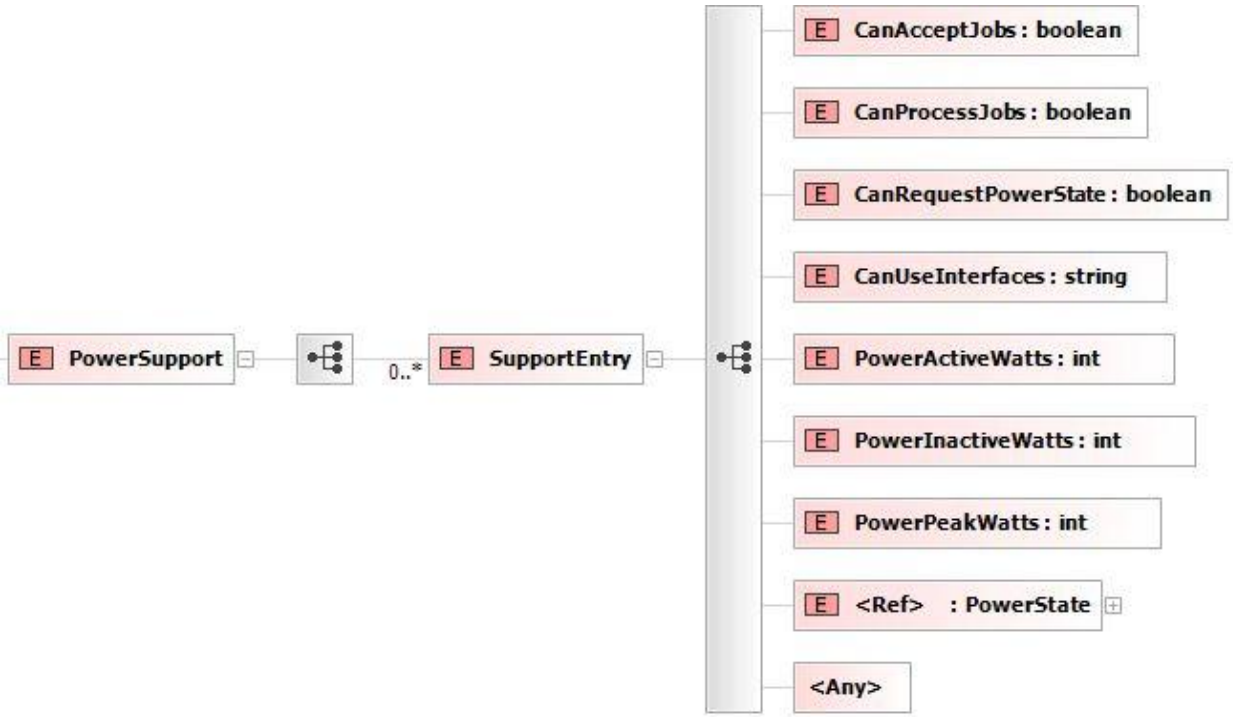

Figure 20 Power Support Schema

Table 8 Power Support Elements

Element	DataType	Description or *Keyword Group*	Reference (PWG Power Management Model) [PWG 5106.4]
Support Entry	Complex		
CanAcceptJobs	boolean		Para 6.1.5 [PWG 5106.4]
CanProcessJobs	boolean		Para 6.1.6 [PWG 5106.4]
CanRequestPowerState	boolean		Para 6.1.7 [PWG 5106.4]
CanUseInterfaces	string	list of interfaces that can be used see 3.1.1.4	Para 6.1.8 [PWG 5106.4]
PowerActiveWatts	int		Para 6.1.3 [PWG 5106.4]
PowerInactiveWatts	int		Para 6.1.2 [PWG 5106.4]
PowerPeakWatts	int		Patra 6.1.4 [PWG 5106.4]
PowerState	keyword	*PowerStateWKV*	Para 6.1.1. [PWG 5106.4]
Any	various	Extensions for Power SupportEntry	

3.1.1.5 Power Transition

This complex Element identifies the design value of the transition time between one stable power state and another. As shown in Figure 21 and listed in Table 9, it is constitued of three Elements: the Starting Power State, the Ending Powerstate and the Transition Time.

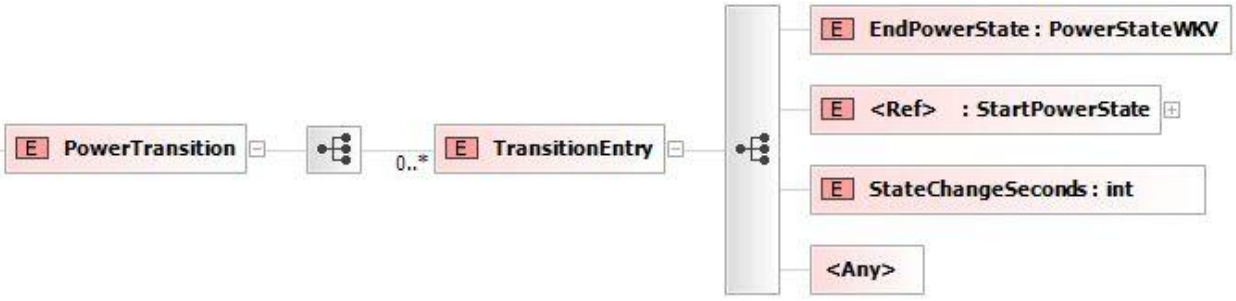

Figure 21 Power Transition Schema

Table 9 Power Transition Elements

Element	DataType	Description or *Keyword Group*	Reference (PWG Power Management Model [PWG 5106.4]
TransitionEntry	Complex		
EndPowerState	keyword	*PowerStateWKV*	Para 6.2.1
StartPowerState	keyword	*PowerStateWKV*	Para 6.2.2
StateChangeSeconds	int	Time in Seconds	Para 6.2.3
Any	various	Extensions to Power TransitionEntry	

3.1.2 Common Subunit Description Elements

The common description Elements for Subunits are identified in Figure 22 and Table 10. Following the convention, Description Elements are those which can be set. As with the common Subunit Status Elements, many of the common Subunit description Elements are power related, but these are settable power Elements. The Elements contained under Subunit Description are described in the following paragraphs.

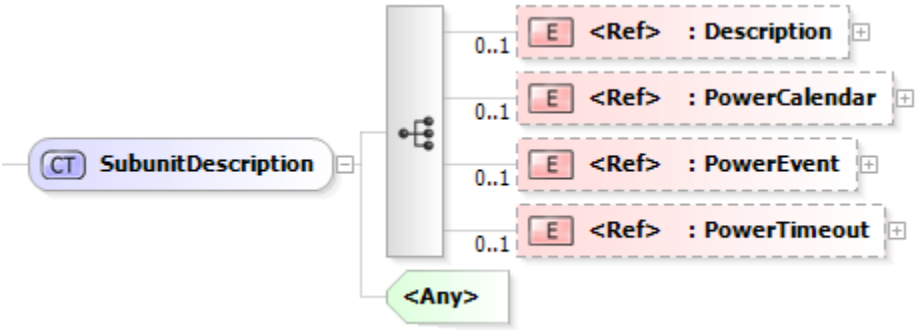

Figure 22 Subunit Description Schema

Table 10 Description of Subunit Description Elements

Element	DataType	Description or *Keyword Group*	Reference
Description	string	textual description of the Subunit, possibly identifying manufacturer, revision and serial number	hrDeviceDescr (hrDeviceTable of [RFC2790]
PowerCalendar	complex	See 3.1.2.1	
PowerEvent	complex	See 3.1.2.2	
PowerTimeout	complex	See 3.1.2.3	

3.1.2.1 Power Calendar

The Power Calendar complex Element identifies configured calendar policies for the Subunit. Power Calendar is represented in Figure 23 and its constituent Elements identified in Table 11. The values of constituent Elements must be persistent across power cycles and hardware reconfigurations.

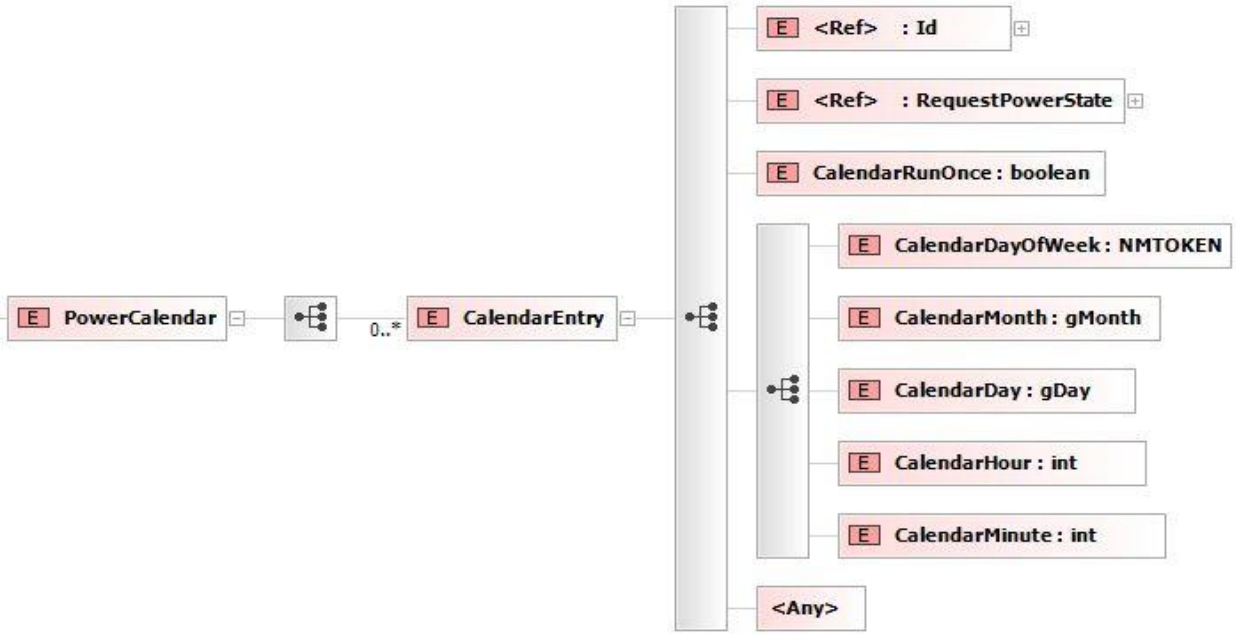

Figure 23 Power Calendar Schema

Table 11 Power Calendar Elements

Element	DataType	Description or Keyword Group	Reference (PWG Power Management Model [PWG 5106.4]
CalendarEntry	Complex		
Id	int	Id of entry, policy	Para 7.3.1 [PWG 5106.4]
RequestPowerState	keyword	*PowerStateWKV*	Para 7.3.2 [PWG 5106.4]
CalendarRunOnce	boolean	Once vs recurring	Para 7.3.3. [PWG 5106.4]
CalendarDayOfWeek	keyword	Requested time of transition to indicated power state	Para 7.3.4 [PWG 5106.4]
CalendarMonth	keyword		Para 7.3.5 [PWG 5106.4]
CalendarDay	keyword		Para 7.3.6 [PWG 5106.4]
CalendarHour	int		Patra 7.3.7 [PWG 5106.4]
CalendarMinute	int		Para 7.3.8 [PWG 5106.4]
Any	Various	Extensions to CalendarEntry	

3.1.2.2 Power Event

The Power Event complex type defines administrator set policies requesting that the Subunit be placed in a specified power state on the occurrence of a specified event. For example, a Marker Subunit might be put in a Standby power state when the required media stock is determined to be not accessible. The Power Event complex Element is represented in Figure 24 and the constituent Elements of this complex Element are identified in Table 12.

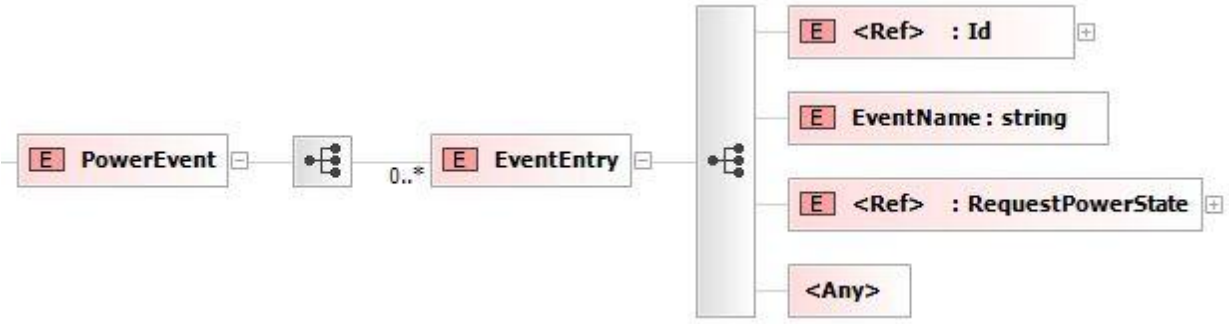

Figure 24 Power Event Schema

Table 12 Power Event Elements

Element	DataType	Description or *Keyword Group*	Reference (PWG Power Management Model) [PWG 5106.4]
EventEntry	complex		
id	int	Number of power event policy	Para 7.4.1 [PWG 5106.4]
EventName	string	See 3.1.2.2, above	Para 7.4.2 [PWG 5106.4]
RequestPowerState	keyword	*PowerStateWKV*	Para 7.4.3 [PWG 5106.4]
Any	Various	Extensions to Power EventEntry	

Note that the Event Name Element, identifying the event on which the power state is to be set, must be structured as either:

(a) The exact case-sensitive label (starting with a lowercase character) of an enumerated value in the PrtAlertCodeTC textual convention in the IANA Printer MIB [IANAPRT] (e.g., 'jam'); or

(b) A case-sensitive keyword (starting with an uppercase character) vendor event name (e.g., 'AcmeCrackedCrock').

3.1.2.3 Power Timeout

This complex Element allows for specifying timeout polices by which a Subunit is to revert to a lower power state after some specified period. The complex Element is represented in Figure 25 and the constituent Elements are identified in Table 13.

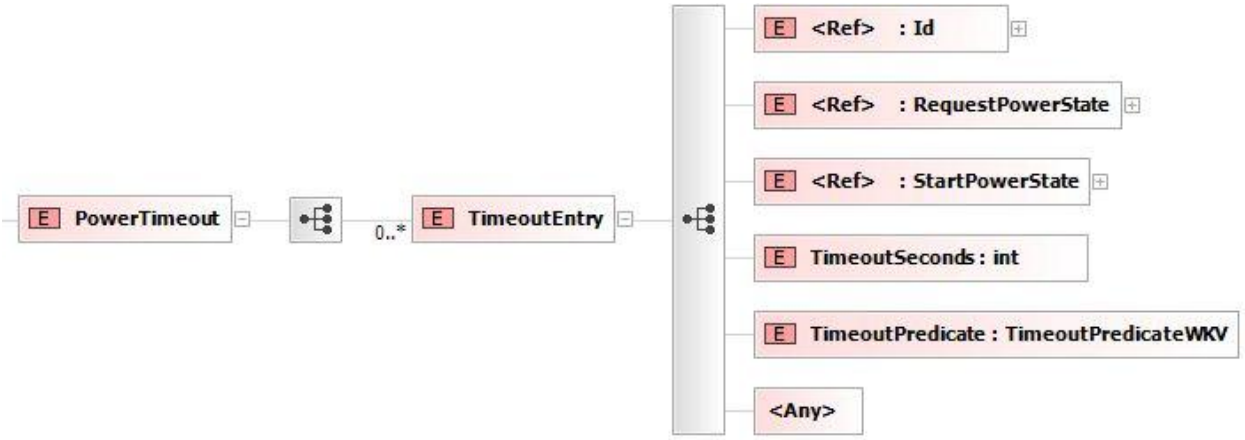

Figure 25 Power Timeout Schema

Table 13 Power Timeout Elements

Element	DataType	Description or Keyword Group	Reference (PWG Power Management Model [PWG 5106.4]
TimeoutEntry	Complex		
Id	int	Id of entry, policy	Para 7.2.1 [PWG 5106.4]
RequestPowerState	keyword	*PowerStateWKV*	Para 7.2.2 [PWG 5106.4]
StartPowerState	keyword	*PowerStateWKV*	Para 7.2.3 [PWG 5106.4]
TimeoutSeconds	int		Para 7.2.5 [PWG 5106.4]
TimeoutPredicate	keyword	*TimeoutPredicateWKV*	Para 7.2.4 [PWG 5106.4]
Any	Various	Extensions to Power TimeoutEntry	

3.2 Consoles

This is a complex Element that represents a user or operator interface panel. It is technically aligned with the prtConsoleGroup [RFC3805]. The constituents of the Console Subunit complex Element are represented in Figure 26 and described in .Table 14.

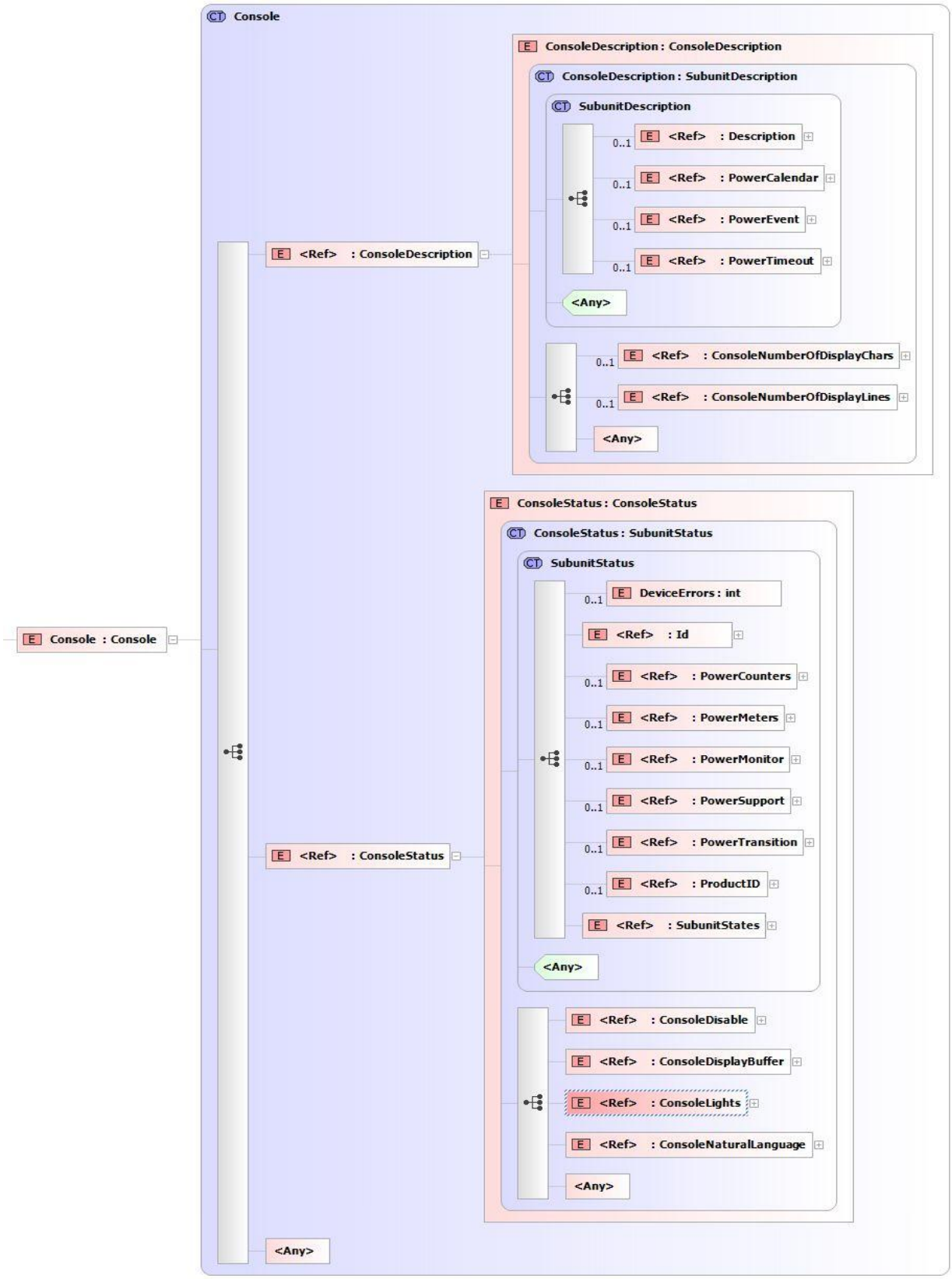

Figure 26 Console Subunit Schema

Table 14 Console Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805]*)
ConsoleDescription	complex		
SubunitDescription	complex	See 3.1.2	
ConsoleNumberOfDisplayChars	int	number of characters per line displayed on the physical display.	prtConsoleNumberOfDisplayChars
ConsoleNumberOfDisplayLines	int	number of lines on the printer's physical display.	prtConsoleNumberOfDisplayLines
Any	various	Extension point for ConsoleDescription	
ConsoleStatus			
SubunitStatus	complex	See 3.1.1	
ConsoleDisable	boolean	input is accepted from the operator console.	prtConsoleDisable
ConsoleDisplayBuffer	complex	Physical display buffer for printer console display or operator panel.	prtConsoleDisplayBufferTable
ConsoleDisplayText	list of string	Contents of buffer associated with each physical line on the display or operator panel	prtConsoleDisplayBufferEntry
ConsoleLights	complex	Set of indicators	
ConsoleLight	list of complex ConsoleLight	description and state information for each light present on the printer console.	prtConsoleLightTable and prtConsoleLightEntry
ConsoleLightOnTime	int	defines the current status of the light in conjunction with ConsoleLightOffTime	prtConsoleOnTime
ConsoleLightOffTime	int	defines the current status of the light in conjunction with ConsoleLightOnTime,.	prtConsoleOffTime
ConsoleLightColor	keyword	color of this light. *ConsoleLightColorWKV*	prtConsoleColor
ConsoleLightInfo	string		prtConsoleDescription
ConsoleNaturalLanguage	string	language, country, and character set to be used for the console.	prtConsoleLocalization and [RFC3066]
Any	various	Extension point for ConsoleStatus	

*Unless otherwise indicated

Figure 27 is a detailed schema of the Console Lights Element. Note that four Elements are available to configure each console light. Console Light Color and Console Light Info are self-explanatory. Console Light On Time and Console Light Off Time define the current status of the light. If both are non-zero, the lamp is blinking and the values presented define the on time and off time, respectively, in milliseconds. If Console Light On Time is zero and ConsoleLightOffTime is non-zero, the lamp is off. If Console Light Off Time is zero and Console Light On Time is non-zero, the lamp is on. If both values are zero, the lamp is off.

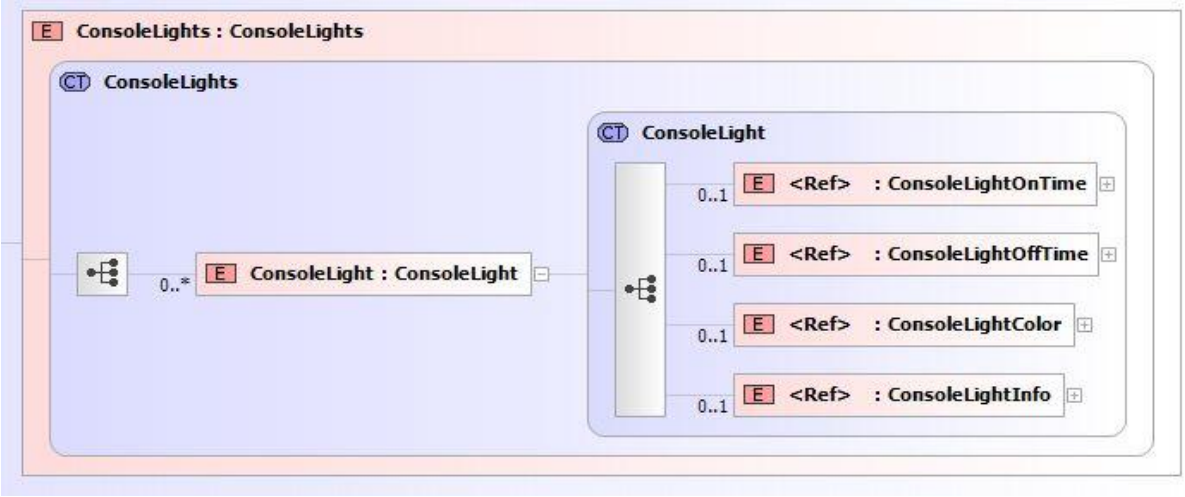

Figure 27 Console Lights Schema

3.3 Covers

This complex Element represents a device's covers, doors, housings or interlocks. It is technically aligned with prtCoverTable [RFC3805]. The constituents of the Cover Subunit complex Element are represented in Figure 28 and described in Table 15.

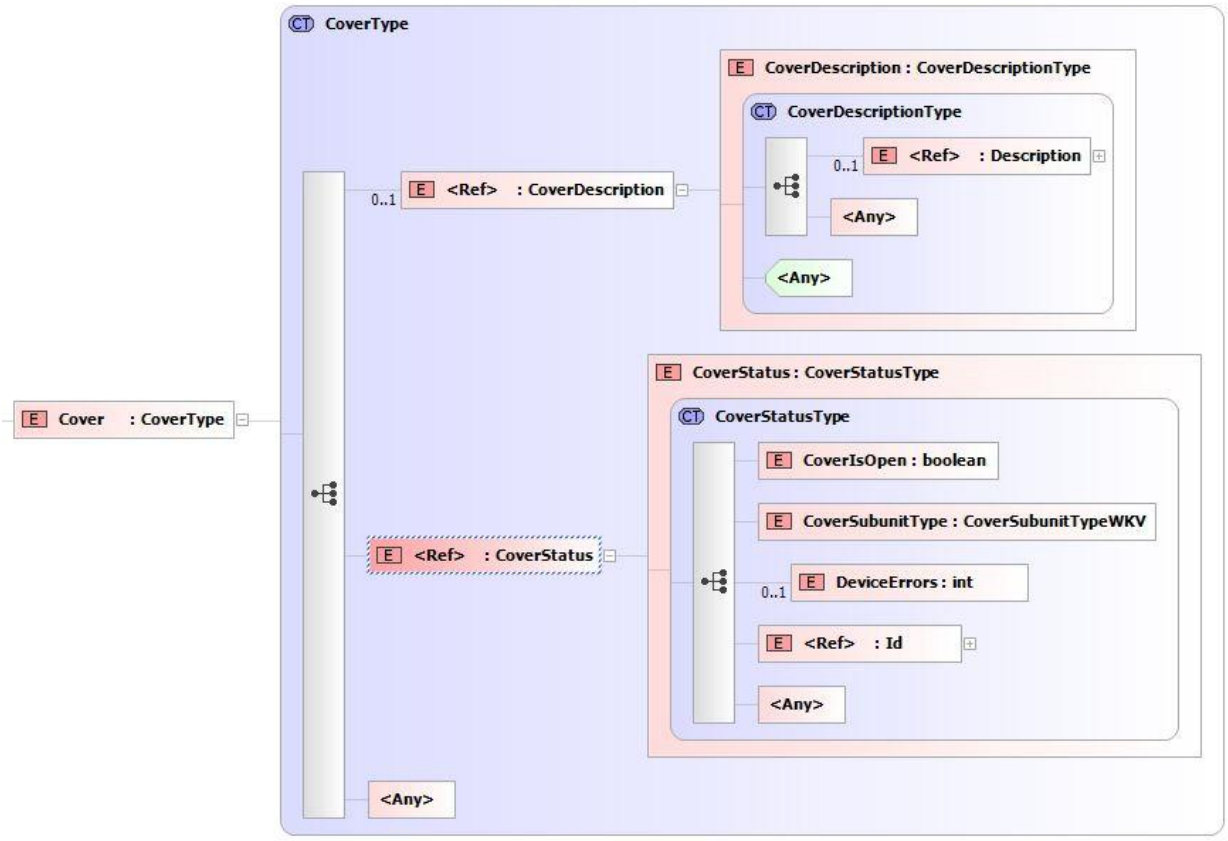

Figure 28 Cover Subunit Schema

Table 15 Cover Elements

Element	DataType	Description or *Keyword Group*	Reference (All [RFC3805])
CoverDescription	complex	The Cover Subunit does not include the Power provisions in the standard Subunit Description and therefore uses a specific CoverDescriptionType.	**[RFC3805]**
Description	string	String description of cover	
Any	various	Extension point for CoverDescription	
CoverStatus			
CoverIsOpen	boolean		
CoverSubunitType	keyword	Differentiates between cover, interlock, etc. *CoverSubunitTypeWKV*	
Any	various	Extension point for CoverStatus	
Any	various	Extension point for Cover	

3.4 FaxModems

This complex Element models the Subunit that provides the interface of the FaxIn and FaxOut Services with the public switched telephone network. It is technically aligned with the mdmMib [RFC1696]. The FaxModem Subunit contains many Elements, with the top structure represented in Figure 29; the FaxModem Description structure represented in Figure 30 and described in Table 16 and the Fax Modem Status structure is represented in Figure 31 and described in Table 17.

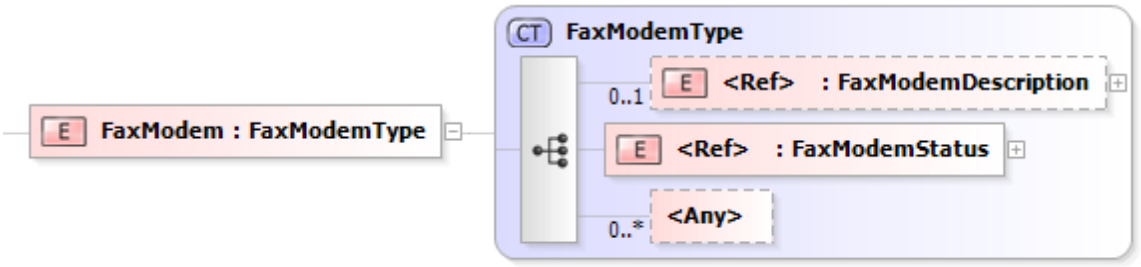

Figure 29 FaxModem

Table 16 Fax Modem Description Elements

Element	DataType	Description or *Keyword Group*	Reference (All[RFC1696])
FaxModemDescription	complex		[RFC1696]
SubunitDescription	complex	See 3.1.2	
AudioVolume	int		
CallSetUpFailTimer	int		
DialingMethod	keyword		
DTEInactivityTimeout	int		
DTESyncAsyncMode	keyword	*DTESynchModeWKV*	
DTESyncTimingSource	keyword	*DTESynchWKV*	
EscapeAction	keyword	*CCEscapeActionWKV*	
FaxModemInfo	string		
FaxModemName	string		
FaxModemModel	string		
FaxModemVersion	string		
FaxModemSerialNumber	string		
FaxModemVendorName	string		
FaxModemLineCapabilities	list of keywords	*LineCapabilityWKV*	
FaxModemDTROnToOff	keyword	*DTRToOffWKV*	
FaxModemDTROffToOn	keyword	*DTRToOnWKV*	
NumberOfRetries	int		
RingsBeforeAnswer	int		
Any	various	Extension point for FaxModemDescription	

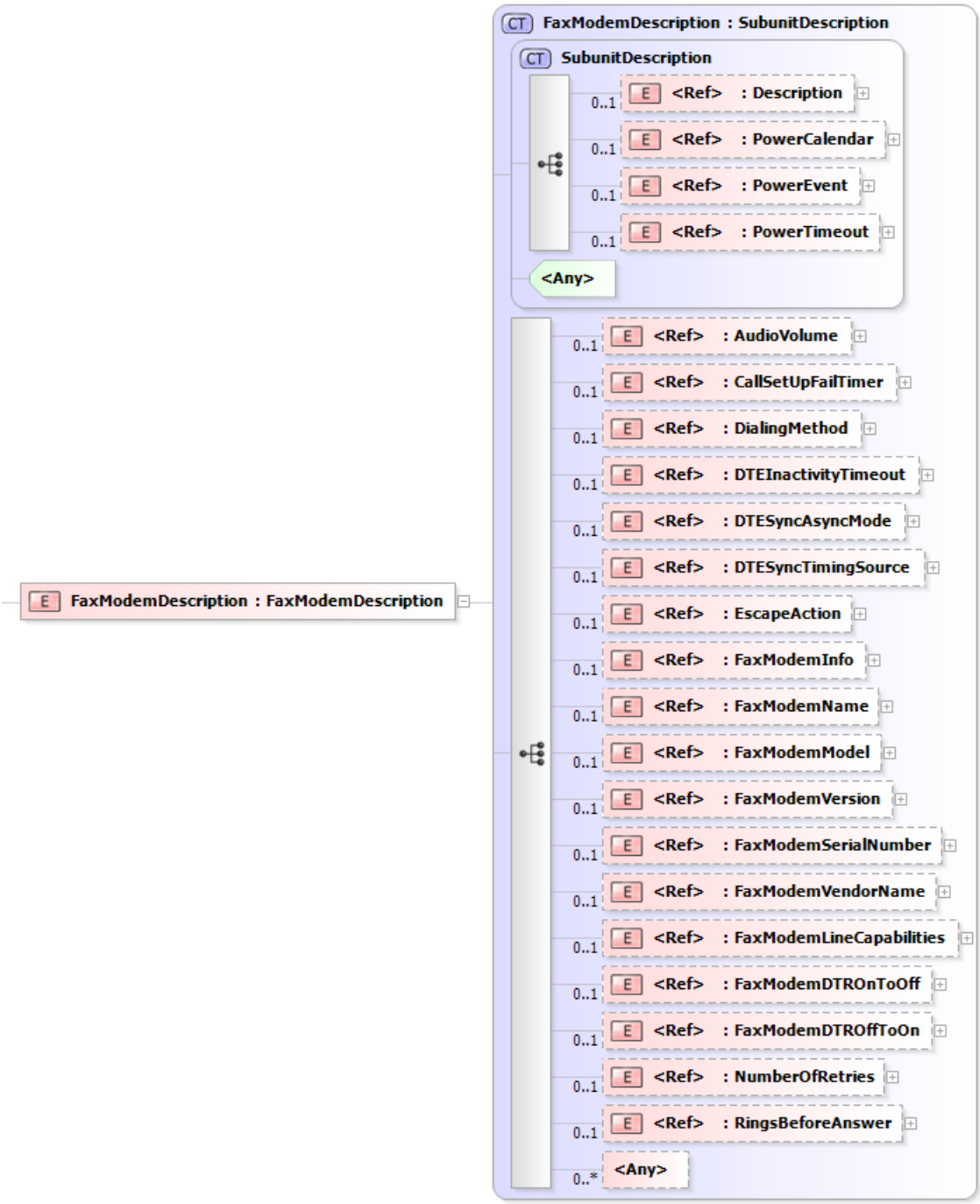

Figure 30 FaxModem Description

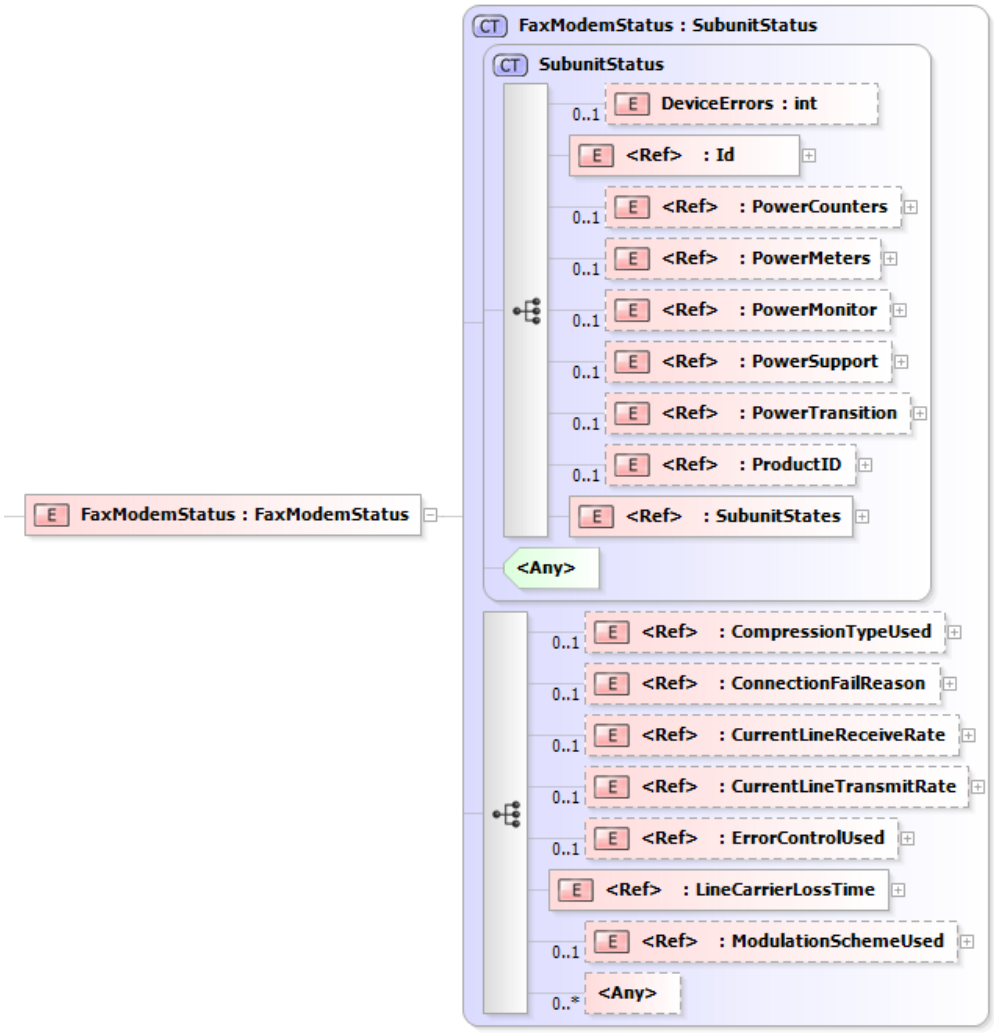

Figure 31 FaxModem Status

Table 17 Fax Modem Status Elements

Element	DataType	Description or *Keyword Group*	Reference (All[RFC1696])
FaxModemStatus	complex		[RFC1696]
SubunitStatus	complex	See 3.1.1	
CompressionTypeUsed	keyword	CompressionType	
ConnectionFailReason	keyword	ConnectionFailReasonWKV	
CurrentLineReceiveRate	int		
CurrentLineTransmitRate	int		
ErrorControlUsed	keyword	ErrorControlWKV	
LineCarrierLossTime	int		
ModulationSchemeUsed	keyword	ModulationSchemeUsed	
Any	various	Extension point for FaxModemStatus	
Any	various	Extension point for FaxModem	

3.5 Finishers

This complex Element represents the devices handling the printed media as it emerges from the Marker Subunit. It is technically aligned with finDevice in the Finisher MIB [RFC3806] and the updates in the IANA Finisher MIB [IANA-FIN]. The constituents of the Finisher Subunit complex Element are represented in Figure 32 and described in Table 18.

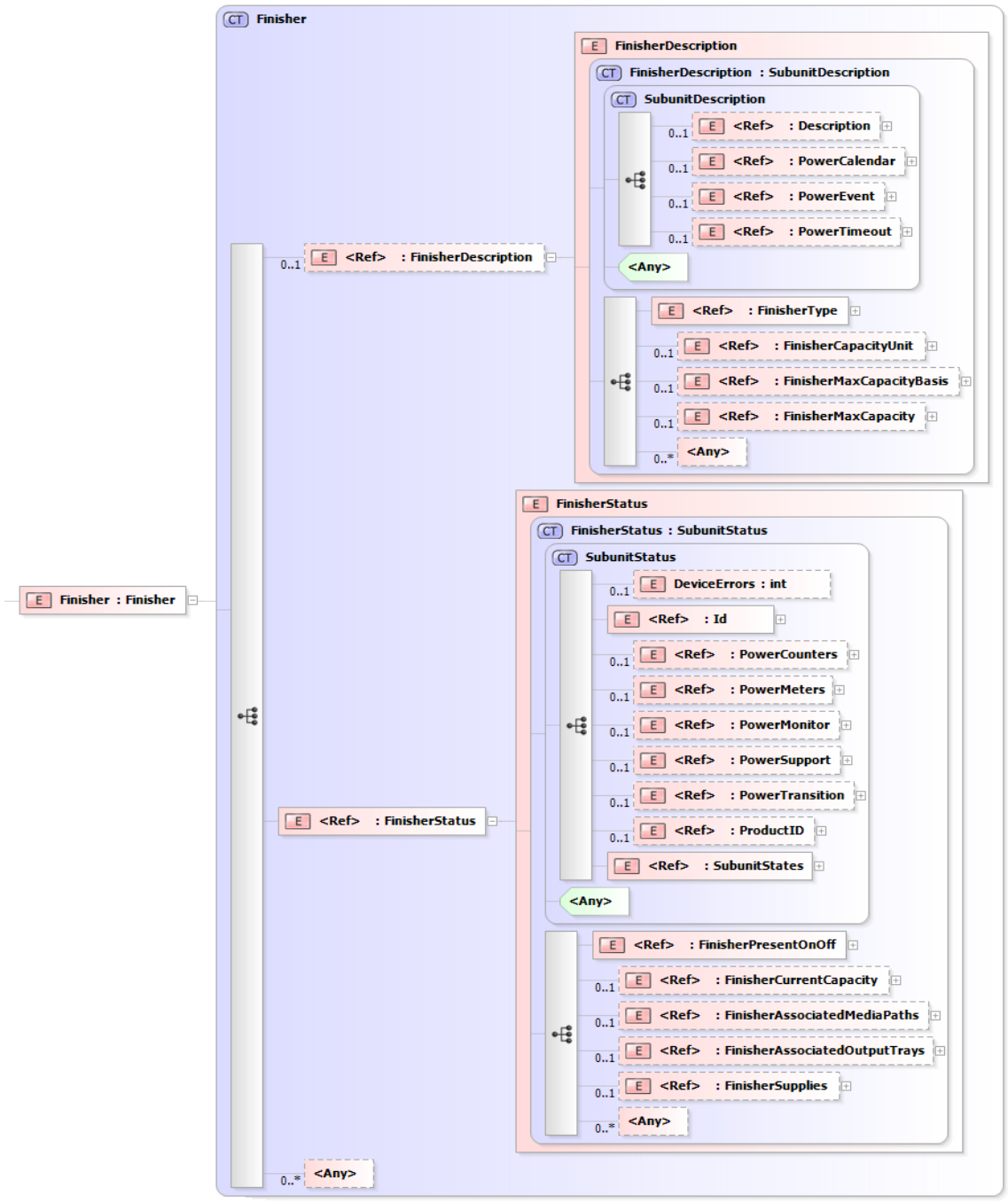

Figure 32 Finisher

Table 18 Finisher Subunit Elements

Element	Datatype	Description or *Keyword Group*	Reference
FinisherDescription	complex		
SubunitDescription	complex	See 3.1.2	
FinisherType	keyword	*FinisherTypeWKV*	FinDeviceTypeTC [RFC3806] updated to [IANA-FIN] and finDeviceType [RFC3806]
FinisherCapacityUnit	keyword	*SubunitCapacityUnitWKV*	finDeviceCapacityUnit [RFC3806] and PrtCapacityUnitTC [RFC3805]
FinisherMaxCapacityBasis	keyword	*ObjectCounterBasisWKV*	finDeviceMaxCapacity [RFC3806]
FinisherMaxCapacity	counter	finisher maximum capacity (in units specified by FinisherCapacityUnit)	finDeviceMaxCapacity [RFC3806]
Any	various	Extension point for FinisherDescription	
FinisherStatus	complex		
SubunitStatus	complex	See 3.1.1	
FinisherPresentOnOff	keyword	*SubunitPresentOnOffWKV*	PresentOnOff [RFC3805] updated to [IANA-PRT] & finDevicePresentOnOff [RFC3806]
FinisherCurrentCapacity	int		finDeviceCurrentCapacity [RFC3806]
FinisherAssociatedMediaPaths	string	finisher associated media paths (bit-mask of values of MediaPathId)	finDeviceAssociatedMediaPaths (bit-mask) [RFC3806]
FinisherAssociatedOutputTrays	string	finisher associated output trays (bit-mask of values of OutputTrayId)	finDeviceAssociatedOutputs (bit-mask) [RFC3806
FinisherSupplies	complex	See Figure 33 and Table 19	
Any	various	Extension point for FinisherStatus	

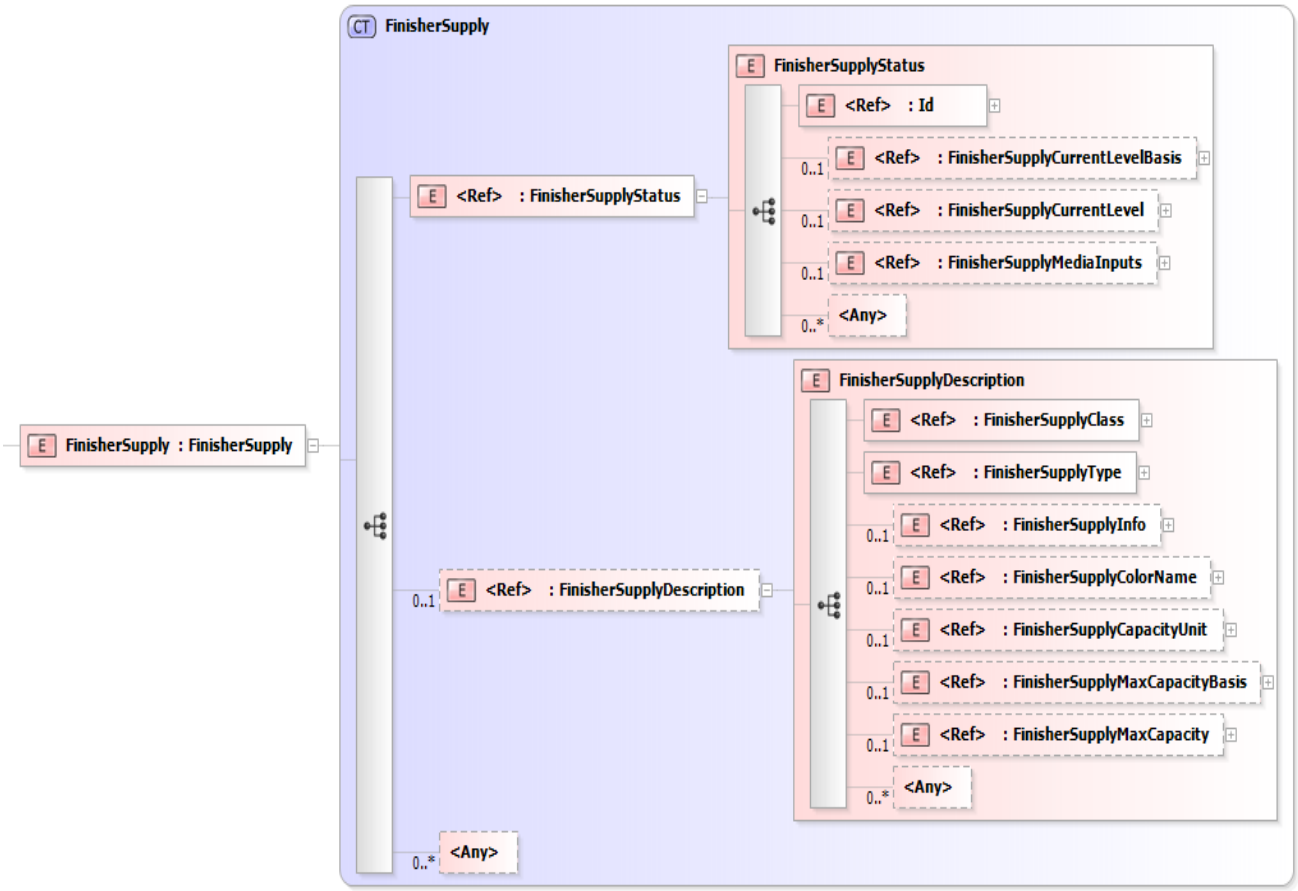

Figure 33 FinisherSupply

Table 19 FinisherSupply Elements

Element	DataType	Description or *Keyword Group*	Reference (RFC3805 and RFC3806)
FinisherSupplyStatus	complex		
Id	int	Index of finisher supplies for finisher	finSupplyIndex in [RFC3806]
FinisherSupplyCurrentLevelBasis	keyword	*ObjectCounterBasisWKV*	finSupplyCurrentLevel [RFC3806]
FinisherSupplyCurrentLevel	int	finisher supply current level (in units specified by FinisherSupplyCapacityUnit)	finSupplyCurrentLevel [RFC3806]
FinisherSupplyMediaInputs	complex	See Figure 34 and Table 20	
Any	various	Extension point for FinisherSupplyStatus	
FinisherSupplyDescription	complex		
FinisherSupplyClass	keyword	*MarkerSupplyClassWKV*	prtMarkerSuppliesClass [RFC3805]
FinisherSupplyType	keyword	*MarkerSupplyTypeWKV*	prtMarkerSuppliesType [RFC3805]
FinisherSupplyInfo	string		prtMarkerSuppliesDescription [RFC3805]
FinisherSupplyColorName	string		prtMarkerSuppliesColorantIndex [RFC3805]
FinisherSupplyCapacityUnit	keyword	*MarkerSupplyCapacityUnitWKV*	prtMarkerSuppliesSupplyUnit [RFC3805]
FinisherSupplyMaxCapacityBasis	keyword	*ObjectCounterBasisWKV*	
FinisherSupplyMaxCapacity	int		prtMarkerSuppliesMaxCapacity [RFC3805]
Any	various	Extension point for FinisherSupplyDescription	

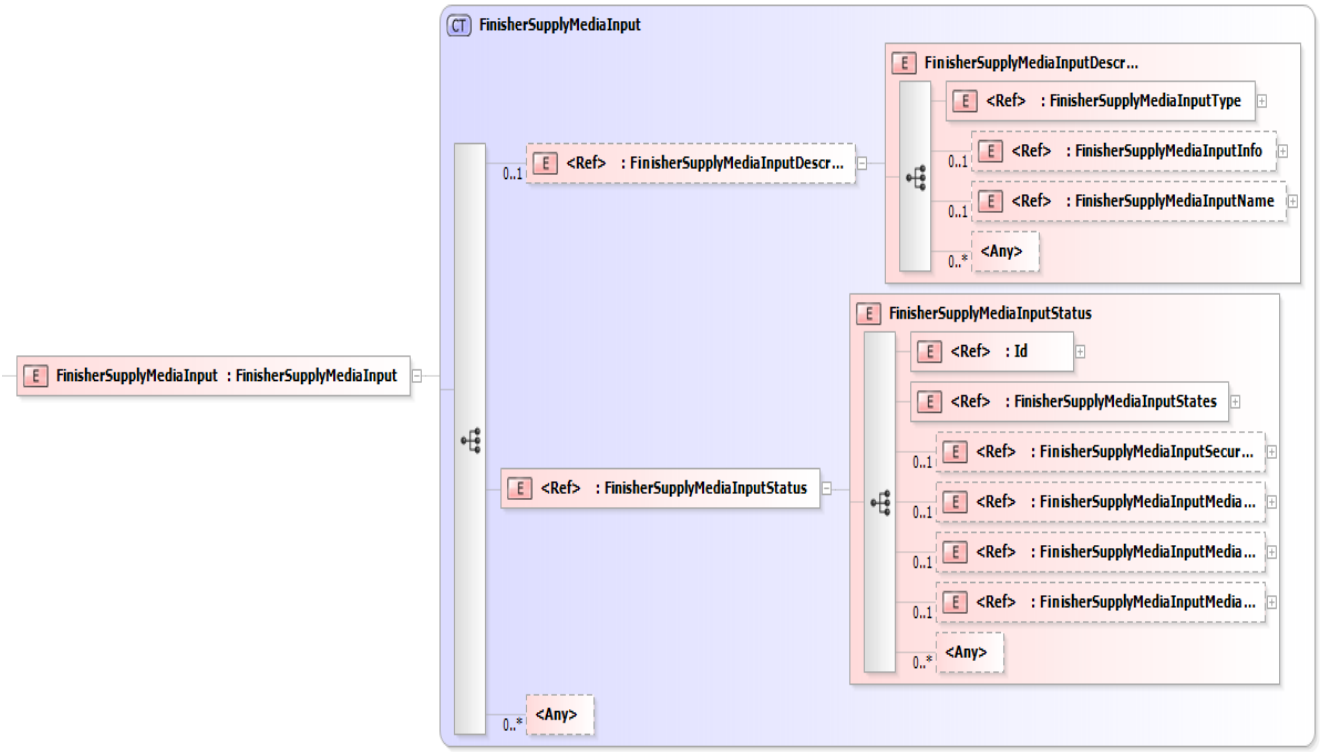

Figure 34 FinisherSupplyMediaInput

Table 20 FinisherSupplyMediaInput Elements

Element	DataType	Description or *Keyword Group*	Reference
FinisherSupplyMediaInputDescription	complex		
FinisherSupplyMediaInputType	keyword	*Input TrayTypeWKV*	PrtInputTypeTC in [RFC3805] & finSupplyMediaInputType [RFC3806]
FinisherSupplyMediaInputInfo	string		finSupplyMediaInputDescription [RFC3806]
FinisherSupplyMediaInputName	string		finSupplyMediaInputName [RFC3806]
Any	various	Extension point for FinisherSupplyMediaInput Description	
FinisherSupplyMediaInputStatus	complex		
Id	int	finisher supply media input local unique integer key	finSupplyMediaInputIndex [RFC3806]
FinisherSupplyMediaInputStates	keyword	*SubunitStateWKV*	
FinisherSupplyMediaInputSecurity	keyword	*SubunitPresentOnOffWKV*	finSupplyMediaInputSecurity (PresentOnOff) [RFC3806]
FinisherSupplyMediaInputMediaSizeName	string	finisher supply media input current media size name	Media Size Self-Describing Names section 5 [PWG5101.1] & finSupplyMediaInputMediaName [RFC3806]
FinisherSupplyMediaInputMediaType	string	finisher supply media input current media type	Media Type Names section 3 in [PWG5101.1] & finSupplyMediaInputMediaType [RFC3806]
FinisherSupplyMediaInputMediaInfo	string	finisher supply media input current media info (weight, etc.)	finSupplyMediaInputMediaWeight & finSupplyMediaInputMediaThickness [RFC3806]
Any	various	Extension point for FinisherSupplyMediaInput Status	

3.6 Input Channels

This complex Element represents the source of Job control information, Resource Requests and Digital Documents input to MFD Services. It is technically aligned with prtChannelTable [RFC3805]. The constituents of the Input Channel Subunit complex Element are represented in Figure 35. Input Channel Elements are identified in Table 21.

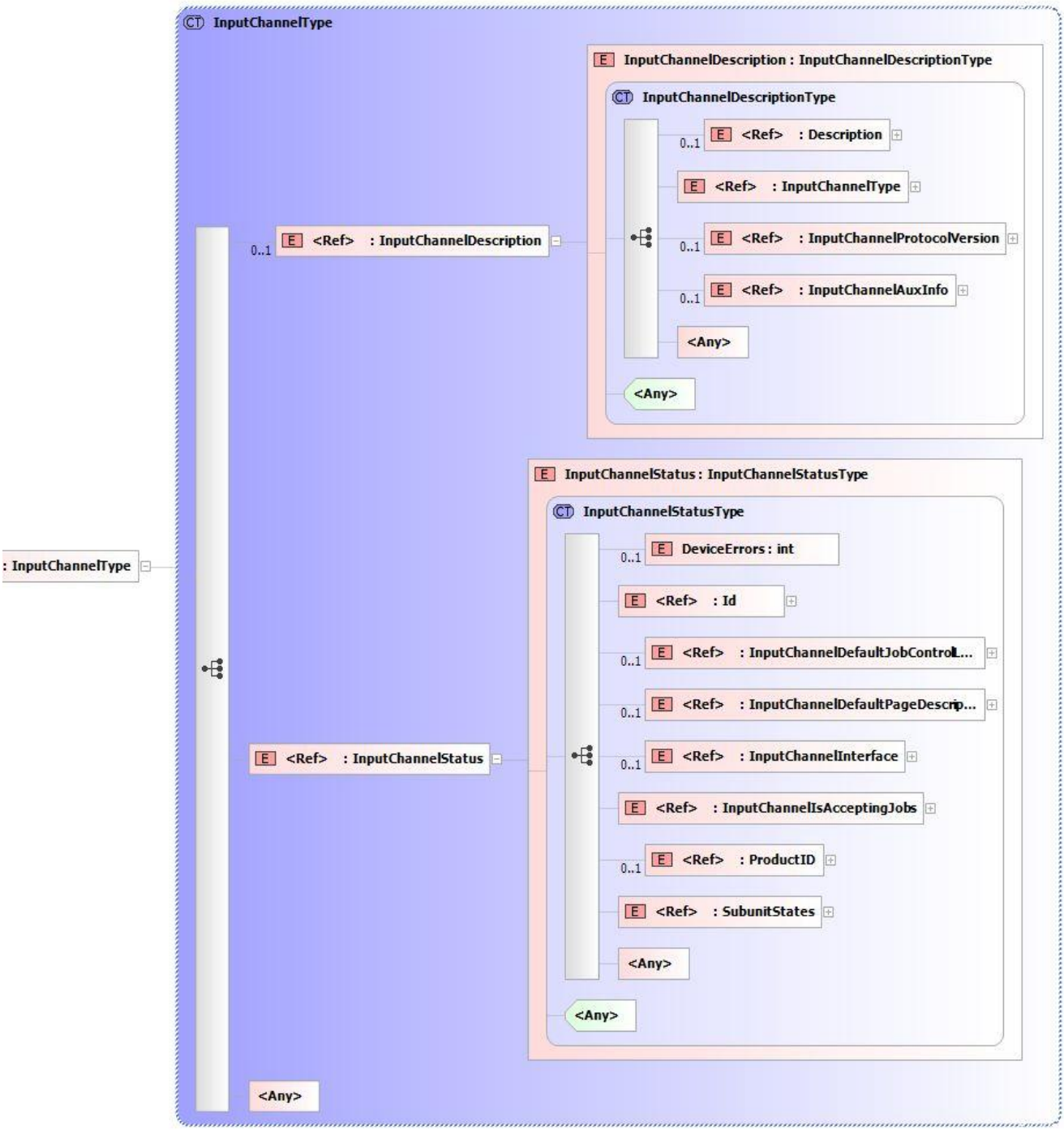

Figure 35 Input Channel Schema

Table 21 Input Channel Elements

Element	Datatype	Description or *Keyword Group*	Reference (all [RFC3805]*)
InputChannelDescription	complex	See Note 1 below.	
Description	string		prtChannelInformation
InputChannelType	keyword	*InputChannelTypeWKV*	PrtChannelTypeTC
InputChannelProtocolVersion	string	input channel protocol version (refines InputChannelType)	prtChannelProtocolVersion
InputChannelAuxInfo	string	input channel auxiliary information in a structured string	PrtChannelTypeTC prtChannelInformation
Any	various	Extension point to InputChannelDescription	
InputChannelStatus	complex	See Note 1 below.	
DeviceErrors	int		hrDeviceErrors[RFC2790]
Id	int		prtChannelIndex
InputChannelDefaultJobControlLanguage	complex	Refers to instance of langage interpreter See Note 2 and Elements of an Interpreter, Table 27	prtChannelCurrentJobCntlLangIndex
InputChannelDefaultPageDescription	complex	Refers to instance of langage interpreter See Note 2 and Elements of an Interpreter, Table 27	prtChannelDefaultPageDescLangIndex
InputChannelInterface	complex	Refers to instance of Interface- See Note 2 and Elements of an Interface, Table 26	prtChannelIfIndex
InputChannelIsAcceptingJobs	boolean	input channel is accepting Jobs (if true)	PrtChannelStateTC prtChannelState
ProductID	string		hrDeviceID [RFC2790]
SubunitStates	keyword	*SubunitStateWKV*	prtChannelStatus
Any	various	Extension point for InputChannelStatus	
Any	various	Extension point to InputChannel	

Notes:

1. Because the Input Channel Subunit is considered software, it does not include the Power Elements in the standard Subunit Description and Subunit Status complex Elements. Therefore, these common Subunit complex Elements are not constituent Elements of the Input Channel Subunit, although some of the constituent Elements of these common complex Elements (such as Product ID) are independently included.

2. The Default Job Control Language, Default Page Description and Interface Elements of the Input Channel Subunit refer to instances of Interpreter and Interface Subunits and are composed of the Elements defined those Subunits. Please see the corresponding Subunit paragraph for the description of these Elements.

*Unless otherwise indicated

3.7 Input Trays

This complex Element represents the mechanisms containing and supplying the Media to be marked in a Marker Subunit. It is technically aligned with prtInputTable [RFC3805]. As shown in Figure 36, the constituents of the Input Tray Subunit complex Element are Input Tray Description, Input Tray Status and Any These Elements are described in Table 22. Input Tray Description (Figure 37) and Input Tray Status (Figure 38) are shown with their Elements described in Table 23 and Table 24.

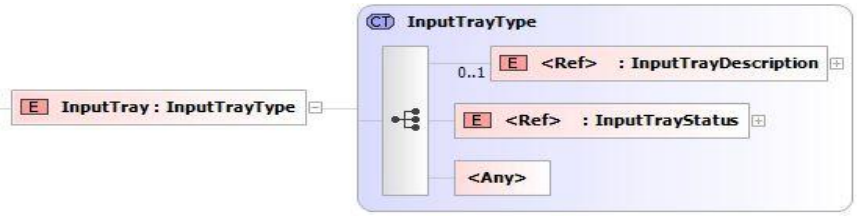

Figure 36 Input Tray

Table 22 Input Trays Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
InputTrayDescription	complex	SeeTable 23	**[RFC3805]**
Input TrayStatus	complex	See Table 24	
Any	various	Extension point to InputTray	

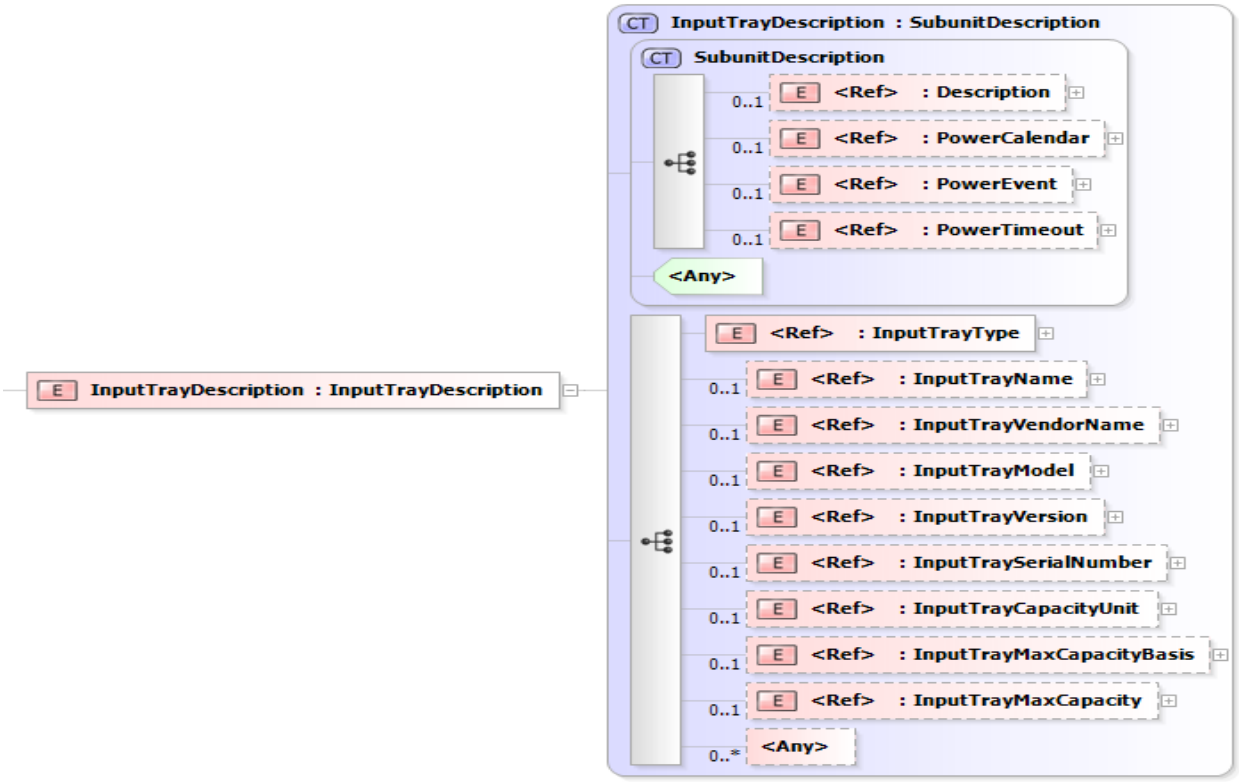

Figure 37 Input Tray Description

Table 23 Input Tray Description Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
InputTrayDescription	complex		
InputTrayType	keyword	*InputTrayTypeWKV*	prtInputType
InputTrayName	string		prtInputName
InputTrayVendorName	string		prtInputVendorName
InputTrayModel	string	input tray /vendor-supplied model of Subunit	prtInputModel
InputTrayVersion	string	input tray vendor-supplied version	prtInputVersion
InputTraySerialNumber	string	input tray vendor-supplied serial number of Subunit	prtInputSerialNumber
InputTrayCapacityUnit	keyword	*SubunitCapacityUnitWKV*	PrtCapacityUnitTC
InputTrayMaxCapacityBasis	keyword	*ObjectCounterBasisWKV*	PrtCapacityUnitTC
InputTrayMaxCapacity	counter	input tray maximum capacity (in units specified by InputTrayCapacityUnit)	PrtCapacityUnitTC
Any	various	Extension point to InputTrayDescription	

Table 24 Input Tray Status Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
InputTrayStatus			
InputTraySecurity	keyword	*SubunitPresentOnOffWKV*	PresentOnOff prtInputSecurity (PresentOnOff)
InputTrayMediaSizeName	string	input tray current media size name - see Media Size Self-Describing Names section 5 in [PWG5101.1]	prtInputMediaName
Input TrayMediaType	string	input tray current media type see Media Type Names section 3 in [PWG5101.1]	prtInputMediaType
Input TrayMediaInfo	string	input tray current media info (weight, color, etc.) see Media Type Names section 3 in [PWG5101.1]	prtInputMediaWeight prtInputMediaColor prtInputMediaFormParts
Input TrayCurrentLevelBasis	keyword	input tray current level basis (i.e., accuracy) *ObjectCounterBasisWKV*	prtInputCurrentLevel
Input TrayCurrentLevel	counter	input tray current level (in units specified by Input TrayCapacityUnit)	prtInputCurrentLevel
Input TrayMediaLoadTimeoutBasis	keyword	*ObjectCounterBasisWKV*	prtInputMediaLoadTimeout
Input TrayMediaLoadTimeout	counter	input tray media load timeout (in seconds)	prtInputMediaLoadTimeout
Input TrayNextInputTrayIdBasis	keyword	*ObjectCounterBasisWKV*	prtInputNextIndex in
Input TrayNextInputTrayId	int	input tray next input tray (after media load timeout) (a value of '0' means 'None', i.e., no input switching)	prtInputNextIndex in
FeedDirection	keyword	For example, ShortEdgeFeed or LongEdgeFeed *FeedOrientationWKV*	
Any	various	Extension point to InputTrayStatus	

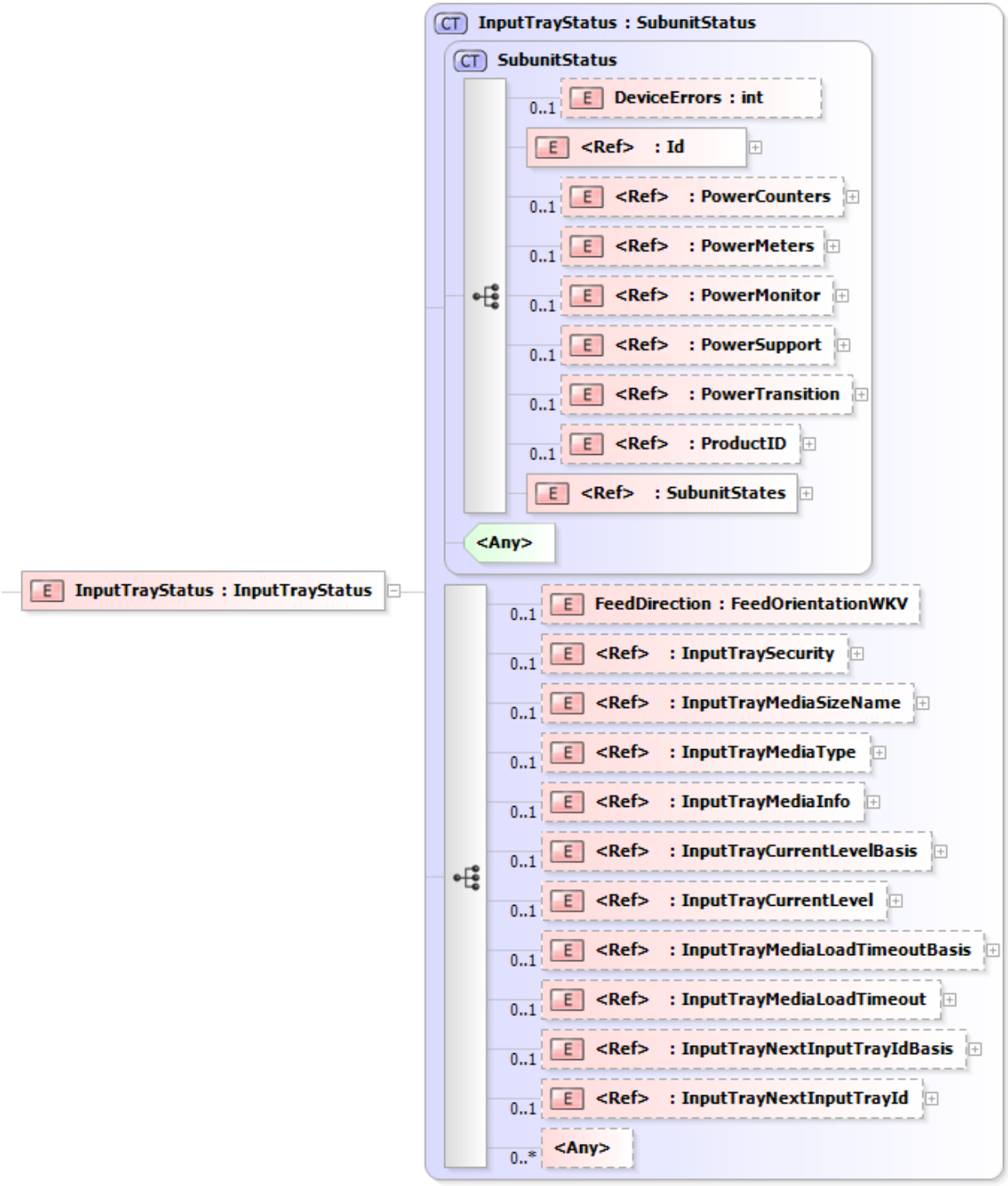

Figure 38 Input Tray Status

3.8 Interfaces

This complex Element represents the communication ports into and out of the device. It is technically aligned with the ifTable defined in MIB-II [RFC2863] The major constituents of the Interface Subunit complex Element are represented in Figure 39; the Interface Description Elements are described in Table 25. Interface Status is modeled in Figure 40 and described in Table 26.

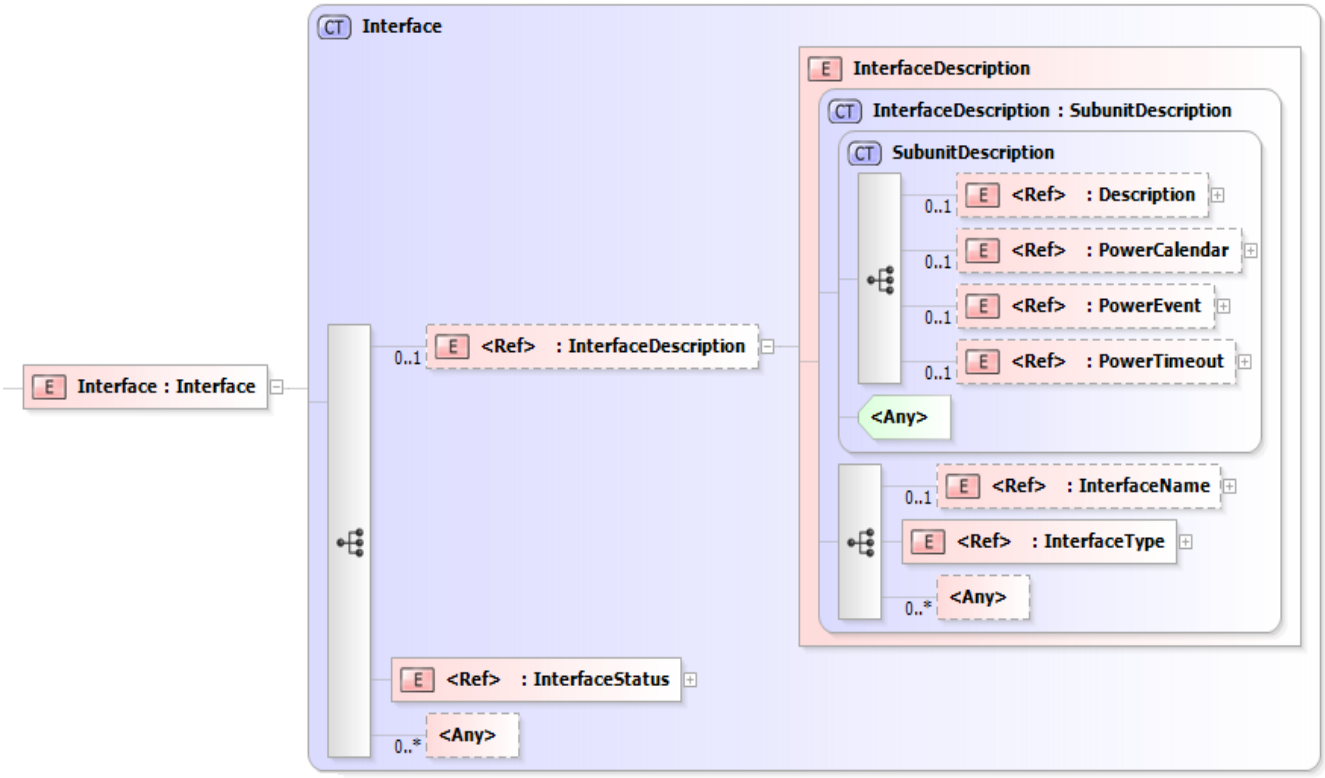

Figure 39 Interface

Table 25 Interface Elements

Element	DataType	Description or *Keyword Group*	Reference(all [RFC2863])*
InterfaceDescription	complex		
SubunitDescription	complex	See 3.1.2	
InterfaceName	string	textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's 'console'.	ifName
InterfaceType	int	type of interface. Values are assigned by the Internet Assigned Numbers Authority IANA, through updating the syntax of the IANAifType textual convention.	ifType and [IANA_NUMBERS]
Any	various	Extension point to InterfaceDescription	
InterfaceStatus	complex	See Table 26	
Any	various	Extension point to Interface	

*Unless otherwise indicated

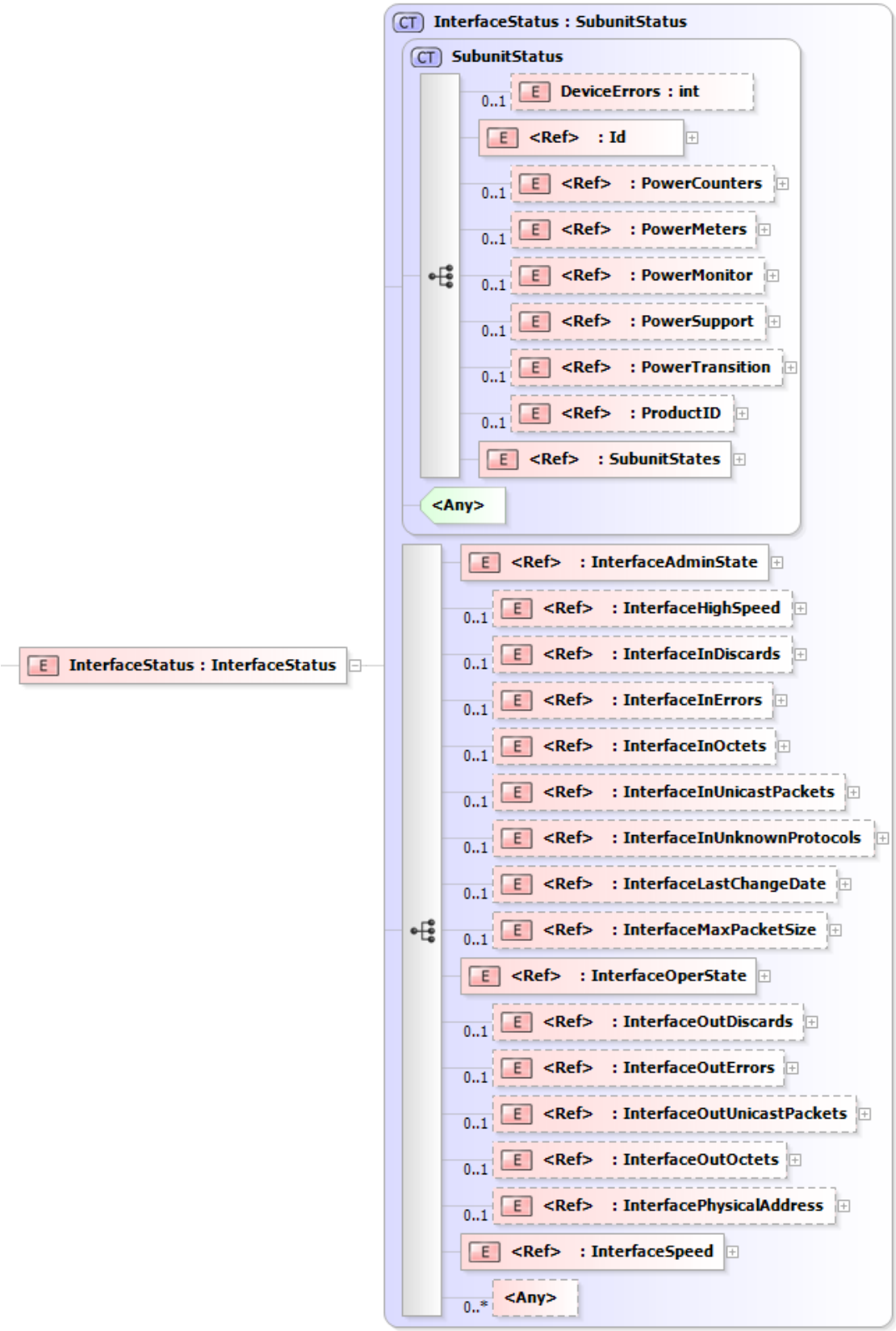

Figure 40 InterfaceStatus

Table 26 InterfaceStatus Elements

Element	DataType	Description or *Keyword Group*	Reference(all [RFC2863])
InterfaceStatus	complex		
SubunitStatus	complex	See 3.1.1	
InterfaceAdminState	keyword	desired state of the interface. *InterfaceAdminStateWKV*	ifAdminStatus
InterfaceHighSpeed	int	estimate of the interface's current bandwidth in units of 1,000,000 bits per second.	ifHighSpeed
InterfaceInDiscards	int	number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol.	ifInDiscards
InterfaceInErrors	int	For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol.	ifInErrors
InterfaceInOctets	int	total number of octets received on the interface, including framing characters.	ifInOctets
InterfaceInUnicastPackets	int	number of packets delivered by this sub-layer to a higher sub-layer which were not addressed to a multicast or broadcast address at this sub-layer.	ifInUcastPkts
InterfaceInUnknownProtocols	int	for packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing, the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol.	ifInUnknownProtos
InterfaceLastChangeDate	dateTime	date/time the interface entered its current operational state NOTE: The syntax of this Element differs from its MIB counterpart int (i.e., seconds)	ifLastChange
InterfaceMaxPacketSize	int	size of the largest packet which can be sent/received on the interface, specified in octets.	ifMtu
InterfaceOperState	keyword	current operational state of the interface. *InterfaceOperStateWKV*	for additional details on ifIndex.
InterfaceOutDiscards	int	number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted.	ifOutDiscards
InterfaceOutErrors	int	for packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors.	ifOutErrors
InterfaceOutUnicastPackets	int	total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.	ifOutUcastPkts
InterfaceOutOctets	int	total number of octets transmitted out of the interface, including framing characters.	ifOutOctets
InterfacePhysicalAddress	string	interface's address at its protocol sub-layer.	ifPhysAddress
InterfaceSpeed	int	estimate of the interface's current bandwidth in bits per second. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value 4,294,967,295 and InterfaceHighSpeed must be used to report the interface's speed.	ifSpeed
Any	various	Extension point to InterfaceStatus	

3.9 Interpreters

This complex Element represents the functions necessary to understand the control languages by which instructions and requests are input and/or to understand the Digital Document description languages by which Digital Documents are input. It is technically aligned with prtInterpreter Table [RFC3805]. The constituents of the Interpreter Subunit complex Element are represented in Figure 41. The Elements are described in Table 27.

Table 27 Interpreter Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
InterpreterDescription	complex	See note below.	
Description	string		
InterpreterLanguageSupported	keyword	control languages which this interpreter can interpret or emulate. The list of keywords is extensible *. InterpreterLanguageFamilyWKV*	prtInterpreterLangFamily
InterpreterLanguageLevel	string	level of the language which this interpreter is interpreting or emulating. This might contain a value like '5e'for an interpreter which is emulating level 5e of the PCL language.	[prtInterpreterLangLevel]
InterpreterLanguageVersion	string	date code or version of the language which this interpreter is interpreting or emulating.	prtInterpreterLangVersion
InterpreterVersion	string	date code, version number, or other product specific information tied to this interpreter. This value is associated with the interpreter, rather than with the version of the language which is being interpreted or emulated.	prtInterpreterVersion
InterpreterLanguageFamily	keyword	family name of the control language which this interpreter is interpreting or emulating. The keyword values are extensible. I*nterpreterLanguageFamilyWKV*	prtInterpreterLangFamily
Any	various	Extension point to InterpreterDescription	
InterpreterStatus		See note below.	
InterpreterDefaultOrientation	keyword	current orientation default for this interpreter. I*nterpreterPageOrientationWKV*	prtInterpreterDefaultOrientation
InterpreterAddressabilityUnit	keyword	units used for Interpreter Addressability. *SubunitAddressabilityUnitWKV*	prtMarkerAddressabilityUnits
InterpreterAddressabilityBasis	keyword	confidence level of the addressability information for the Interpreter. *ObjectCounterBasisWKV*	prtInterpreterFeedAddressability
InterpreterAddressabilityCrossFeed	int	addressability in the cross feed direction i.e., fast scan direction in InterpreterAddressabilityUnits.	prtInterpreterXFeedAddressability
InterpreterAddressabilityFeed	int	addressability in the feed direction i.e., slow – scan direction in InterpreterAddressabilityUnits	prtInterpreterFeedAddressability
InterpreterDefaultCharsetIn	string	default coded character set for input octets encountered outside a context in which the Page Description Language establishes octet interpretation.	prtInterpreterDefaultCharSetIn
InterpreterDefaultCharsetOut	string	default character set for data from interpreter through output channel i.e., the 'backchannel'	prtInterpreterDefaultCharSetOut
InterpreterIsTwoWay	boolean	interpreter returns information to the host.	prtInterpreterTwoWay
Any	various	Extension point to InterpreterStatus	
Any	various	Extension point to Interpreter	

Note: Because the Interpreter Subunit is considered software, the Power elements in the common. Subunit Status and Subunit Description base Elements are not applicable. The Interpreter Subunit therefore does not include these common complex base Elements, although it does include some lower level elements that are also in the common complex Subunit Status and Subunit Description Elements.

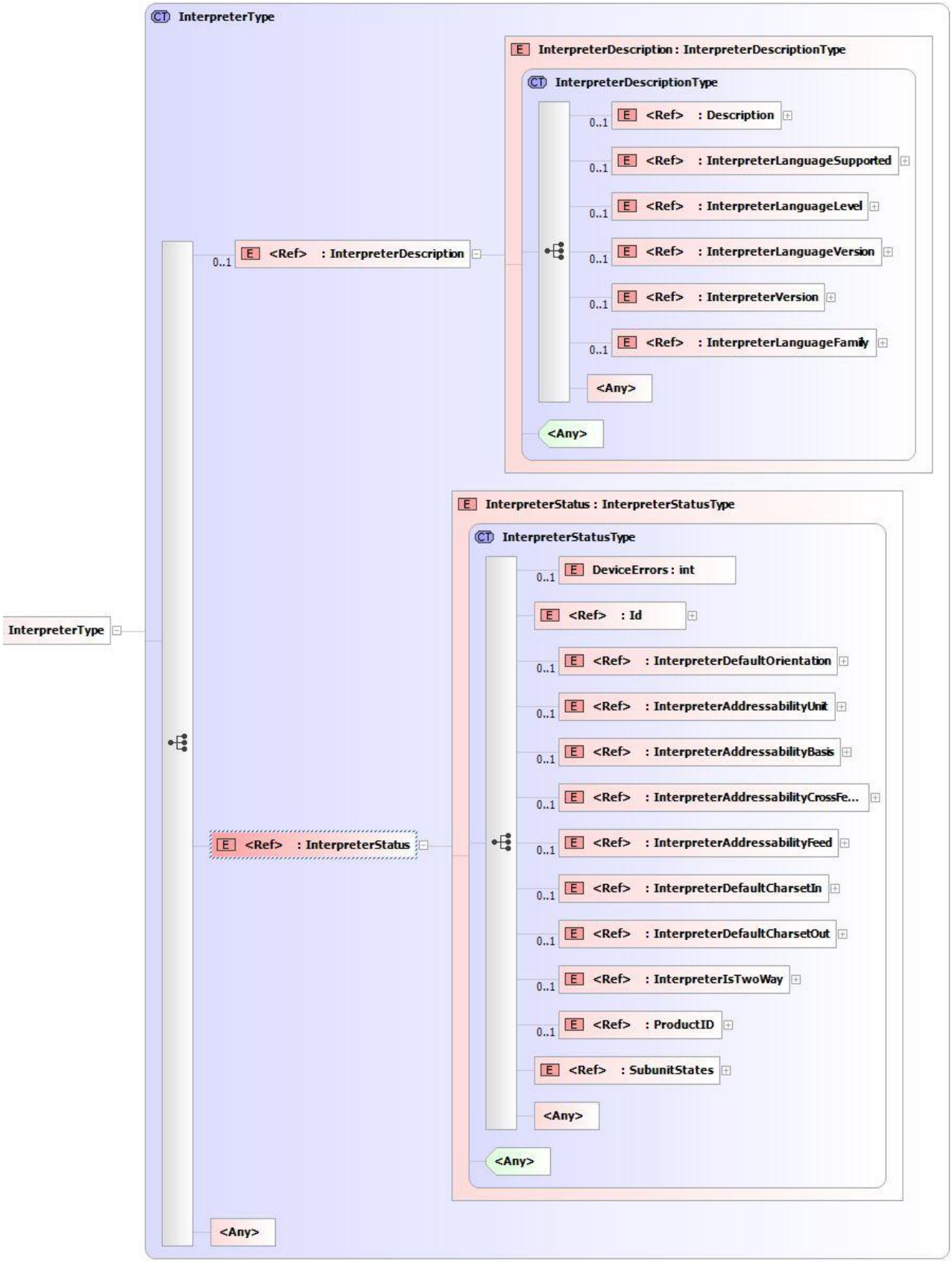

Figure 41 Interpreter

3.10 Marker

This complex Element represents the mechanisms by which marks are impressed upon the media. It is technically aligned with prtMarkerTable [RFC3805] The constituents of the Marker Subunit and Marker Subunit Description complex Elements are represented in Figure 42 and described in Table 28. Marker Status is represented in Figure 43 and Table 29. Two of the constituent Elements of Marker Status, Marker Colorants and Marker Supply are separately represented in Figure 44 and Figure 45, with their Elements described in Table 30 and Table 31.

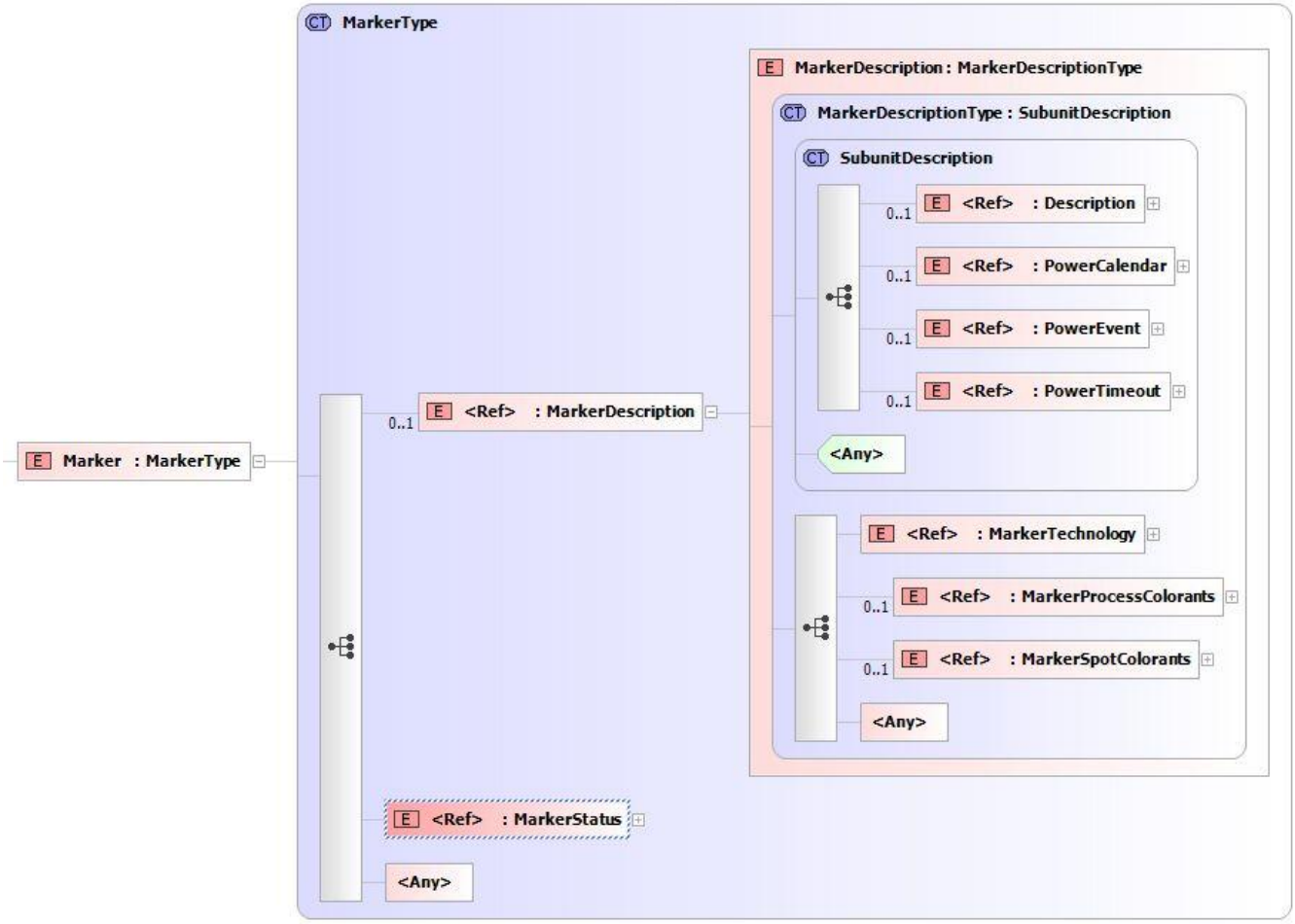

Figure 42 Marker Schema

Table 28 Marker Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
MarkerDescription	complex		
SubunitDescription	complex	See 3.1.2	
MarkerTechnology	keyword	MarkerTechnologyWKV	PrtMarkerMarkTechTC
MarkerProcessColorants	counter		prtMarkerProcessColorants
MarkerSpotColorants	counter		prtMarkerSpotColorants
Any	various	Extension point to MarkerDescription	

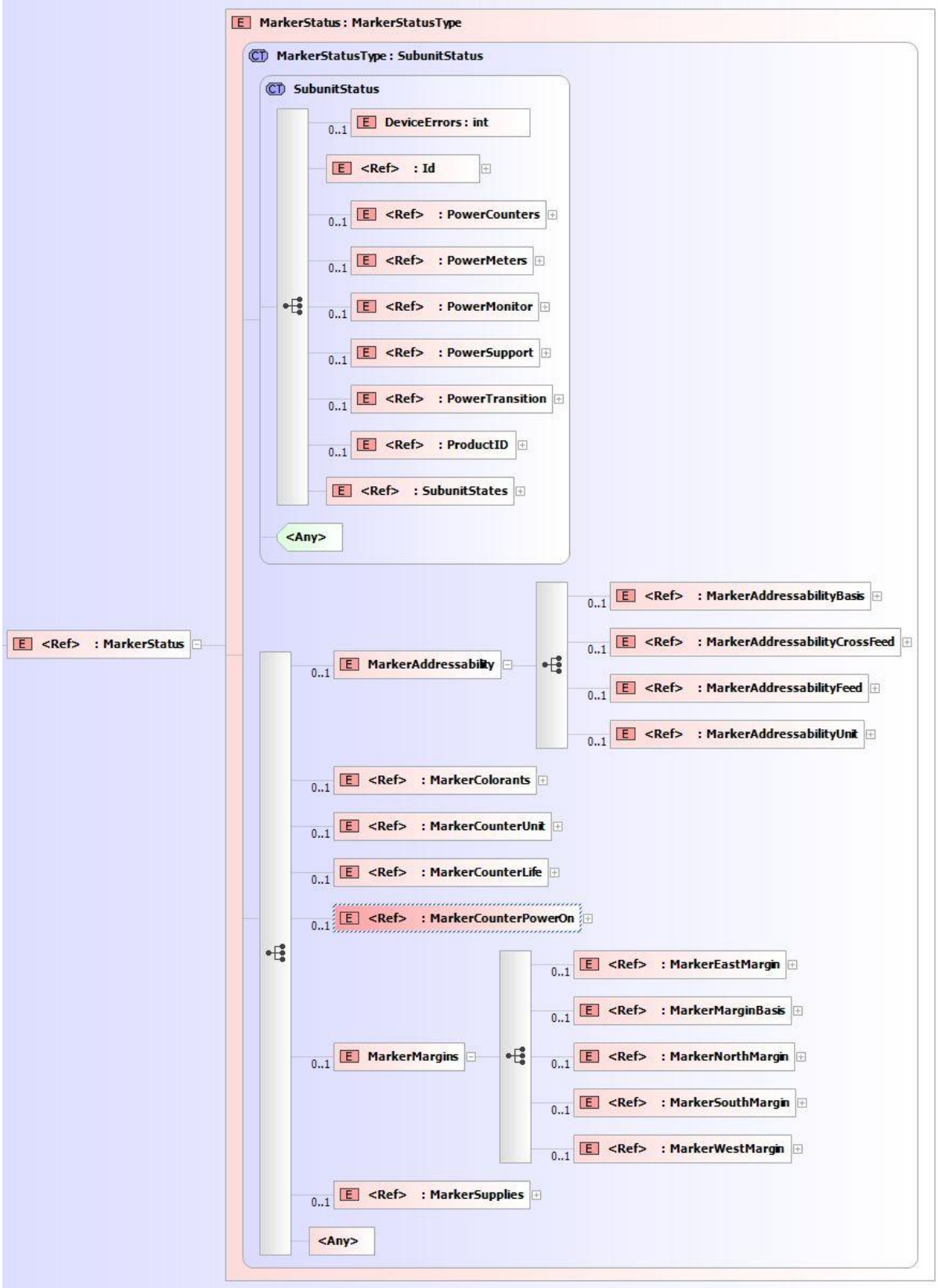

Figure 43 MarkerStatus Schema

Table 29 Marker Status Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
SubunitStatus	complex	See 3.1.1	
MarkerAddressability	complex		
MarkerAddressabilityBasis	keyword	*ObjectCounterBasisWKV*	prtMarkerFeed/xFeedAddressability out of band values
MarkerAddressabilityCrossFeed	counter		prtMarkerXFeedAddressability
MarkerAddressabilityFeed	counter		prtMarkerFeedAddressability
MarkerAddressabilityUnit	keyword	*SubunitAddressabilityUnitWKV*	PrtMarkerAddressabilityUnitTC
MarkerColorants	complex	See Table 30	
MarkerCounterUnit	keyword	*SubunitCounterUnitWKV*	prtMarkerCounterUnit
MarkerCounterLife	counter	marker usage over Subunit lifetime	prtMarkerLifeCount
MarkerCounterPowerOn	counter	marker usage since last power on	prtMarkerPowerOnCount
MarkerMargins	complex		
MarkerEastMargin	counter		prtMarkerEastMargin
MarkerMarginBasis	keyword	*ObjectCounterBasisWKV*	prtMarkerNorth/South/East/WestMargin out of band value
MarkerNorthMargin	counter		prtMarkerNorthMargin
MarkerSouthMargin	counter		prtMarkerSouthMargin
MarkerWestMargin	counter		prtMarkerWestMargin
MarkerSupplies	complex	See Table 31 for Elements defining each marker supply	
Any	various	Extension point for Marker Status	
Any	various	Extension point for Marker	

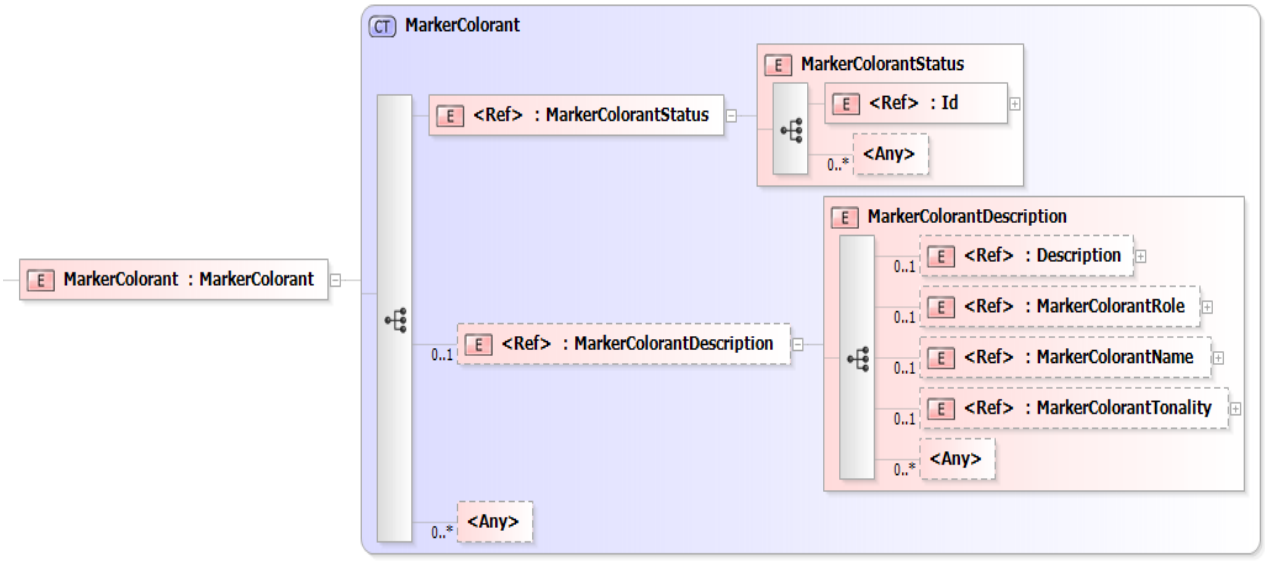

Figure 44 Marker Colorant Schema

Table 30 Market Colorant Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
MarkerColorant	complex		
MarkerColorantStatus	complex		
Id	int		prtMarkerColorantIndex
Any	various	Extension point for MarkerColorantStatus	
MarkerColorantDescription	complex		
MarkerColorantRole	keyword	*MarkerColorantRoleWKV*	prtMarkerColorantRole
MarkerColorantInfo	string	marker colorant vendor-supplied description (in locale specified by SystemNaturalLanguage)	
MarkerColorantName	string	marker colorant standard name see marker colorant vendor-supplied description (in locale specified by SystemNaturalLanguage)	prtMarkerColorantValue
MarkerColorantTonality	counter	marker colorant distinct levels of tonality (levels of tonal difference available for rendering)	prtMarkerColorantTonality
Any	various	Extension point for MarkerColorantDescription	
Any	various	Extension point for MarkerColorant	

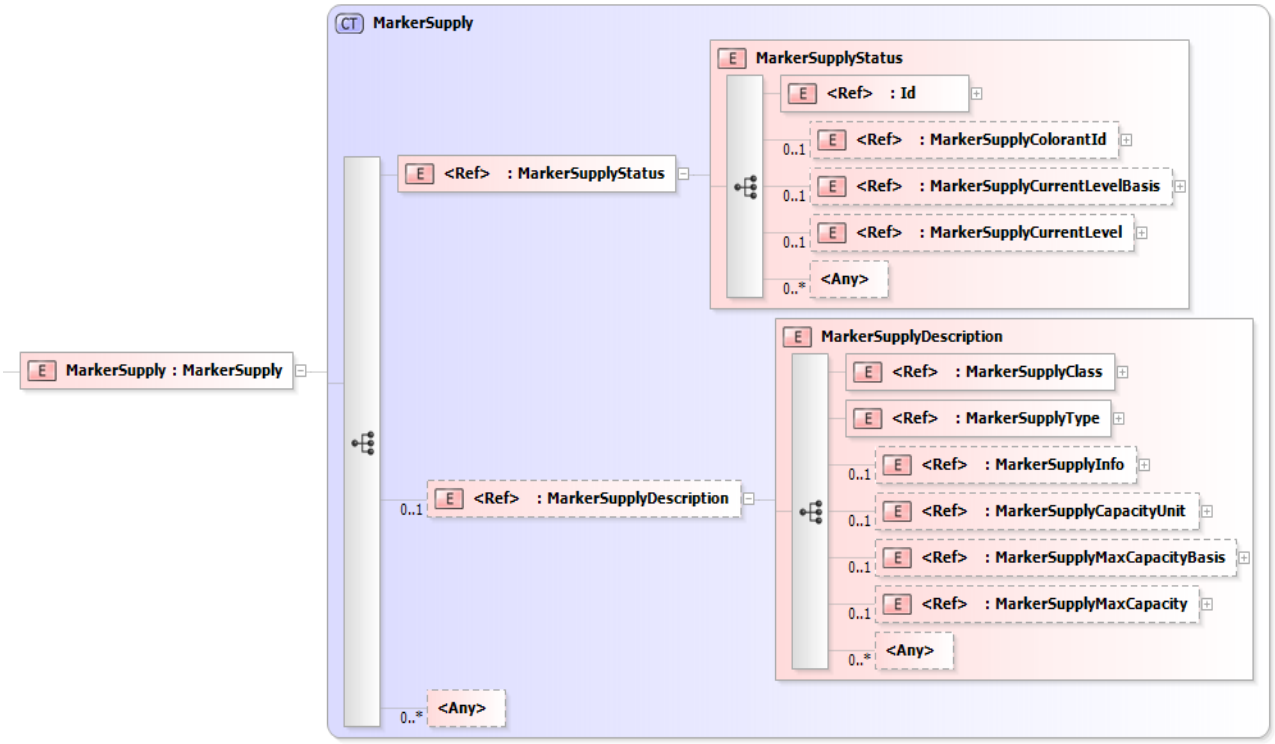

Figure 45 Marker Supply

Table 31 Marker Supply Elements

Element	DataType	Description or Keyword Group	Reference (all [RFC3805])
MarkerSupplyStatus	complex		
MarkerSupplyId	int		prtMarkerSuppliesIndex
MarkerSupplyColorantId	pointer		prtMarkerSuppliesColorantIndex
MarkerSupplyCurrentLevelBasis	keyword	*ObjectCounterBasisWKV*	prtMarkerSuppliesLevel
MarkerSupplyCurrentLevel	counter		prtMarkerSuppliesLevel
Any	various	Extension point for MarkerSupplyStatus	
MarkerSupplyDescription	complex		
MarkerSupplyClass	keyword	*MarkerSupplyClassWKV*	prtMarkerSuppliesClass
MarkerSupplyType	keyword	*MarkerSupplyTypeWKV*	prtMarkerSuppliesType
MarkerSupplyInfo	string		prtMarkerSuppliesDescription
MarkerSupplyCapacityUnit	keyword	*MarkerSupplyCapacityUnitWKV*	prtMarkerSuppliesSupplyUnit
MarkerSupplyMaxCapacityBasis	keyword	*ObjectCounterBasisWKV*	prtMarkerSuppliesMaxCapacity
MarkerSupplyMaxCapacity	counter		prtMarkerSuppliesMaxCapacity
Any	various	Extension point for MarkerSupplyDescription	
Any	various	Extension point for MarkerSupply	

3.11 Media Paths

This is more fully identified as the Print Media Paths This complex Element represents the mechanisms by which the media is moved through a Printer Subunit, from Input Tray to Output Tray or Finisher. It is technically aligned with prtMediaPathTable [RFC3805]. The Elements of a Media Path Subunit are represented in Figure 46. The constituent Elements are described in Table 32. Note that Scan Media Paths are a separate Subunit described in paragraph3.15.

Table 32 Media Paths Elements

Element	DataType	Description or Keyword Group	Reference (all [RFC3805])
MediaPathDescription	complex		
SubunitDescription	complex	See 3.1.2	
MediaPathMaxSpeed	counter		prtMediaPathMaxSpeed
MediaPathMaxSpeedUnit	keyword	PrtMediaPathMaxSpeedPrintUnitTC in [RFC3805] *MediaPathMaxSpeedUnitWKV*	prtMediaPathMaxSpeedPrintUnit
MediaPathMaxMediaSizeName	string	Media Size Self-Describing Names section 5 in [PWG5101.1]	prtMediaPathMediaSizeUnit; prtMediaPathMaxMediaFeedDir prtMediaPathMaxMediaXFeedDir
MediaPathMinMediaSizeName	string		prtMediaPathMediaSizeUnit; prtMediaPathMinMediaFeedDir prtMediaPathMinMediaXFeedDir
MediaPathType	keyword	PrtMediaPathTypeTC in [RFC3805] *MediaPathTypeWKV;*	prtMediaPathType
MediaTypesSupported	list of keywords	Sequence of allowed mediatype values *MediaTypeWKV*	
Any	various	Extension point for MediaPathDescription	
MediaPathStatus	complex		
SubunitStatus	complex	See 3.1.1	prtMediaPathStatus
Any	various	Extension point for MediaPathStatus	
Any	various	Extension point MediaPath	

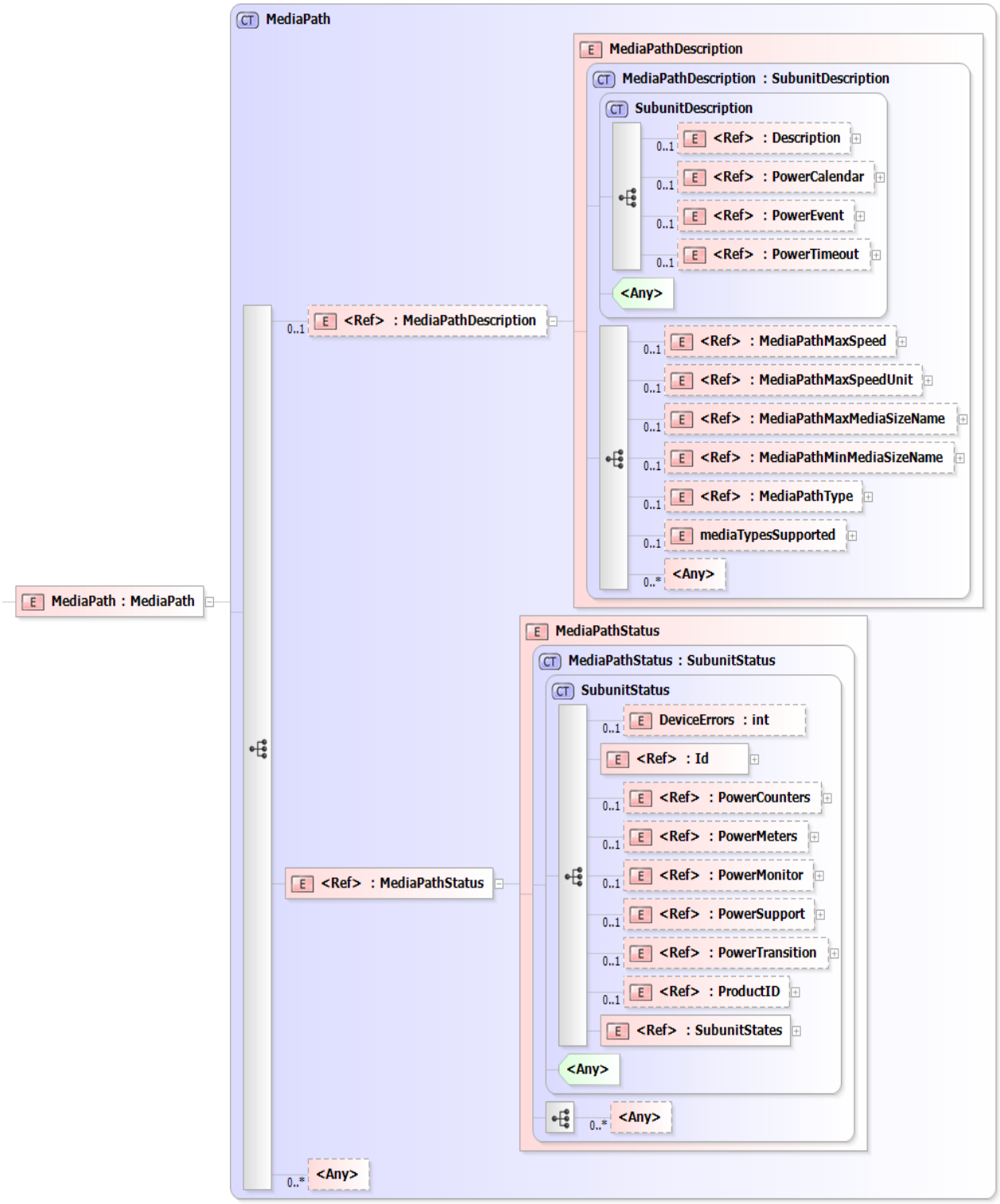

Figure 46 Media Path

3.12 Output Channels

This complex Element represents the destination pathways for Digital Documents. It is technically aligned with prtChannelTable [RFC3805]. Output Channels are associated with an underlying Interface and are usually bound to a transport protocol or file system. The constituents of this complex Element are represented in Figure 47; the Elements are described in Table 33.

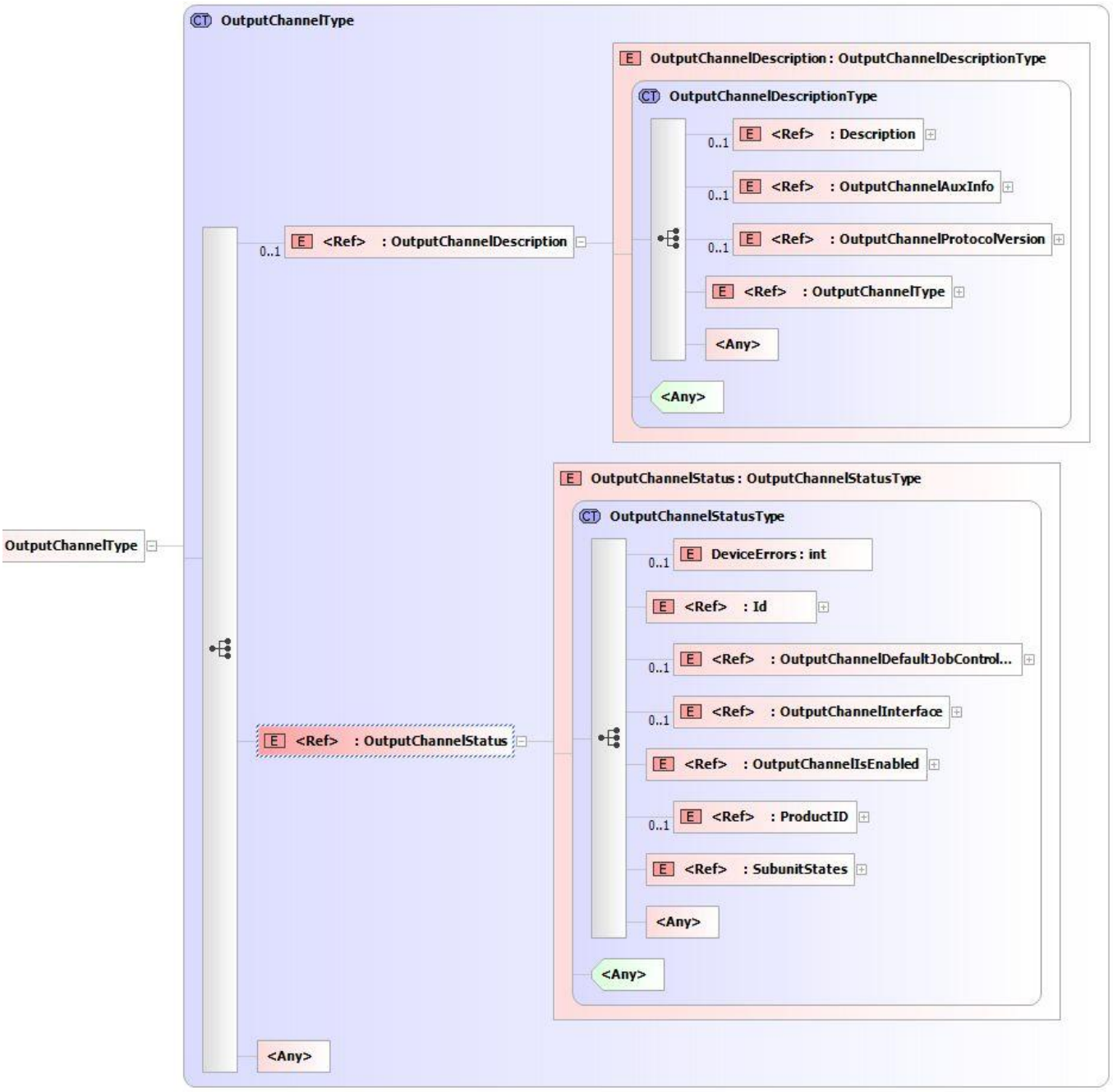

Figure 47 Output Channel

Table 33 Output Channel Elements

Element	DataType	Description or *Keyword Group*	Reference [RFC3805]*
OutputChannelDescription	complex	See Note below.	
Description	string		prtChannelInformation
OutputChannelAuxInfo	string	additional protocol information needed to use the Output Channel's protocol. The information is protocol specific.	ifDescr [RFC1213]
OutputChannelProtocolVersion	string	version of the protocol associated with an Output Channel instance in a protocol specific manner.	prtChannelProtocolVersion
OutputChannelType	keyword	This Element specifies the protocol associated with an Output Channel instance. *OutputChannelTypeWKV*	prtChannelType
Any	various	extension point for OutputChannelDescription	
OutputChannelStatus	complex	See Note below.	
DeviceErrors	int		hrDeviceErrors [RFC2790]
Id	int		
OutputChannelDefault JobControlLanguage	complex	the Job Control Language i.e., Interpreter associated with the channel. This defines the syntax used for Document control functions See 3.9 for constituent Elements	prtChannelDefaultPageDescLangIndex
OutputChannelInterface	complex	This is the Interface instance associated with this instance of an Output Channel. See 3.8 for constituent Elements.	prtChannelIfIndex
OutputChannelIsEnabled	boolean	indicates whether or not an Output Channel instance is available for	PrtChannelStateTC prtChannelState
Product Id	string	Human readable name	hrDeviceID [RFC2790]
SubunitStates	keyword	SubunitStateWKV	prtChannelStatus
Any	various	Extension point for OutputChannelStatus	
Any	various	Extension point for OutputChannel	

Note: Because the Output Channel Subunit is considered software, the Power elements in the common SubunitStatus and Subunit Description are not applicable. The Output Channel Subunit therefore does not include these common complex base Elements, although it does include some lower level elements that are also in the common complex Subunit Status and Subunit Description Elements.

*Unless otherwise indicated

3.13 Output Trays

This complex Element represents the trays or bins capable of receiving Hard Copy Documents that have been printed or scanned. The definition of Output Trays is technically aligned with prtOutputTable [RFC3805]. The basic constituents of the Output Tray Subunit complex Element are represented in Figure 48, with Output Tray Description and Output Tray Status being pictured in Figure 49 and Figure 50 respectively and described in Table 34.

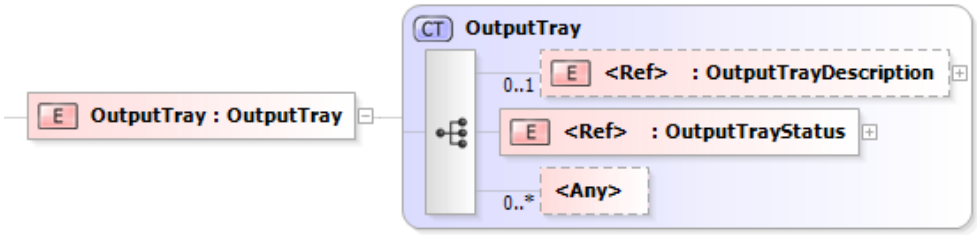

Figure 48 Output Tray

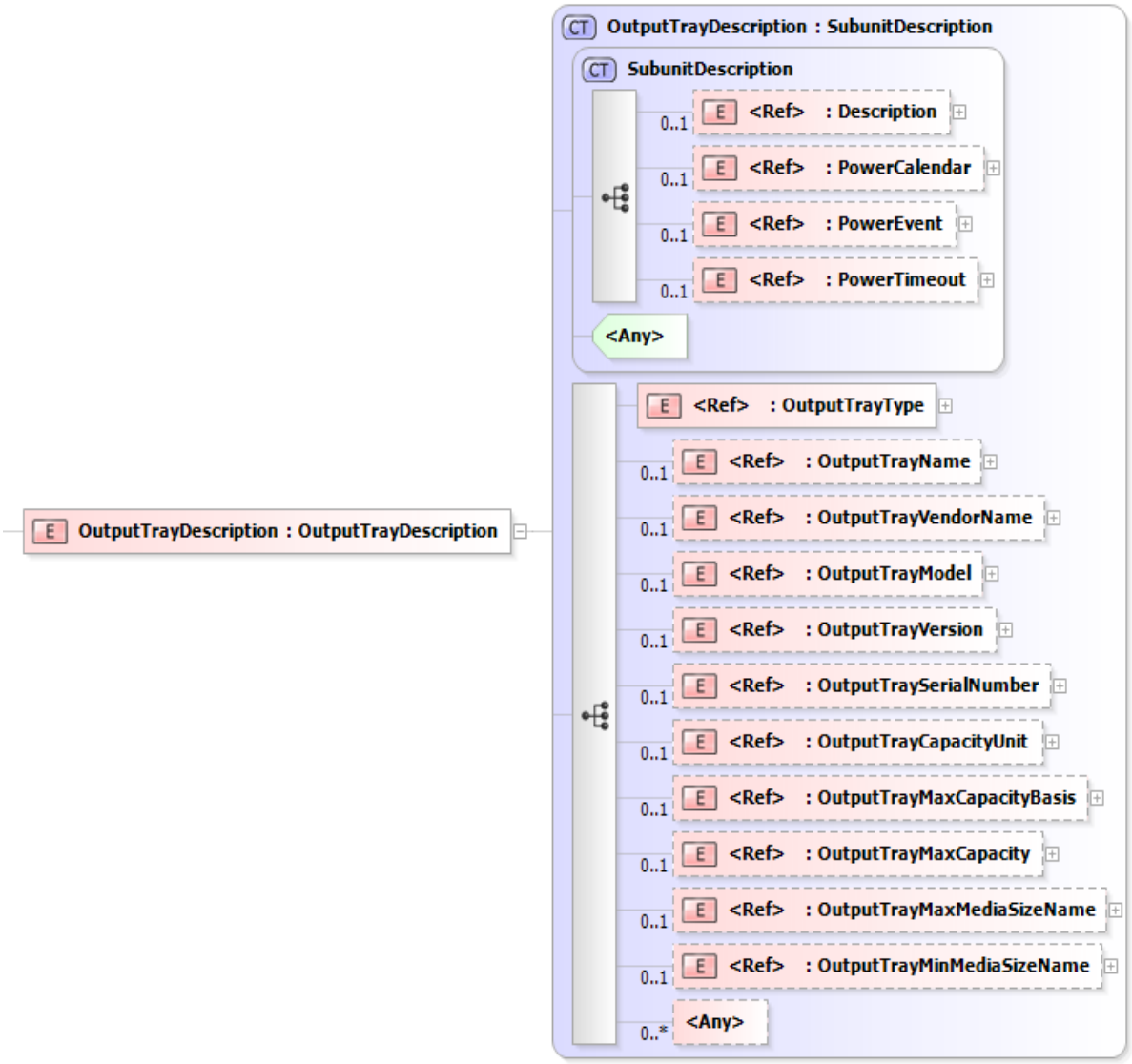

Figure 49 Output Tray Description

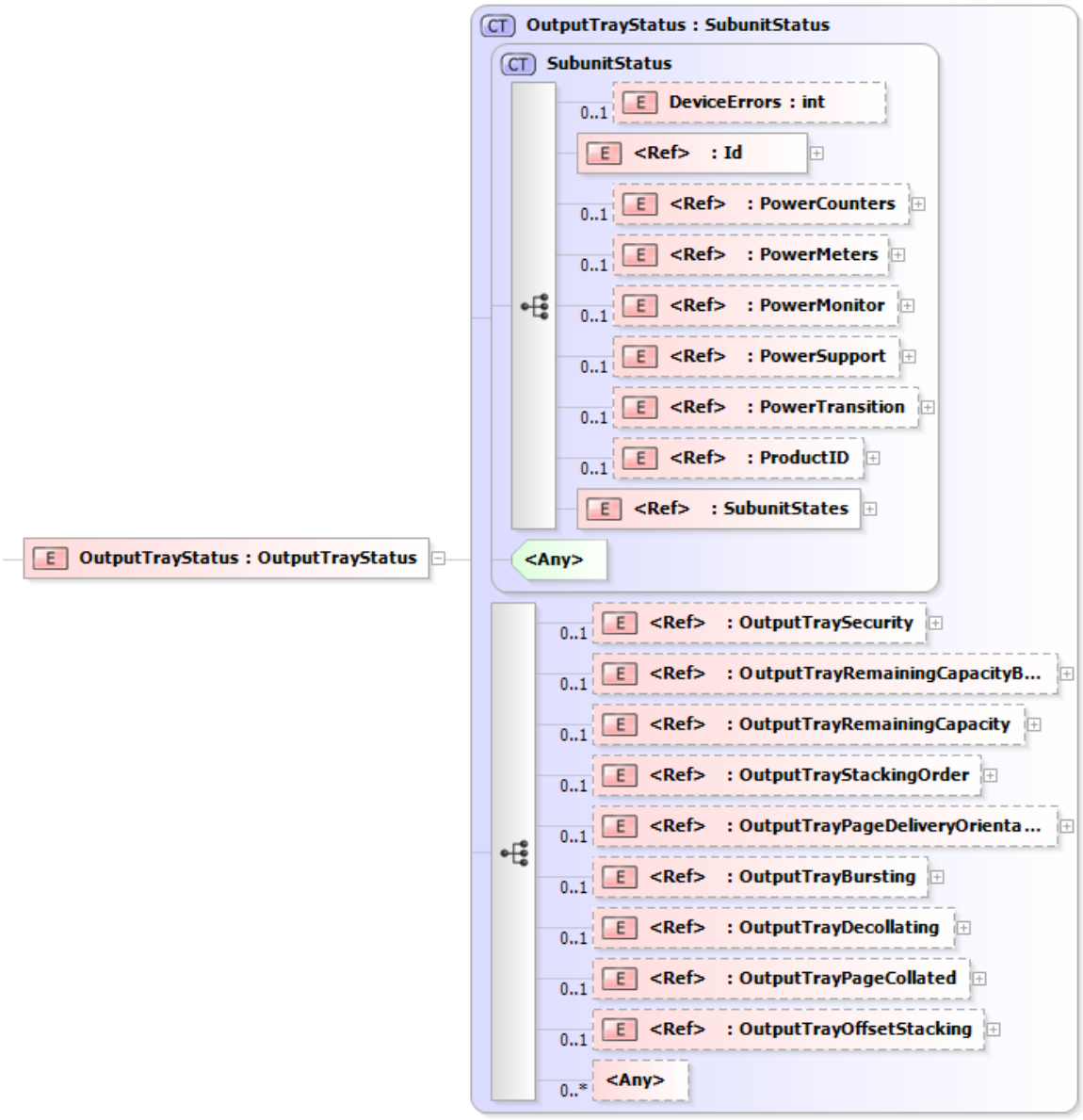

Figure 50 Output Tray Status

Table 34 Output Tray Elements

Element	DataType	Description or *Keyword Group*	Reference (all [RFC3805])
OutputTrayDescription	complex		
SubunitDescription	complex	See 3.1.2	
OutputTrayType	keyword	PrtOutputTypeTC in [RFC3805] *OutputTrayTypeWKV;*	prtOutputType
OutputTrayName	string		prtOutputName
OutputTrayVendorName	string		prtOutputVendorName
OutputTrayModel	string		prtOutputModel
OutputTrayVersion	string		prtOutputVersion
OutputTraySerialNumber	string		prtOutputSerialNumber
OutputTrayCapacityUnit	keyword	PrtCapacityUnitTC in [RFC3805] *SubunitCapacityUnitWKV;*	prtOutputCapacityUnit
OutputTrayMaxCapacityBasis	keyword	ObjectCounterBasisWKV	prtOutputMaxCapacity
OutputTrayMaxCapacity	counter	output tray maximum capacity (in units specified by OutputTrayCapacityUnit	
OutputTrayMaxMediaSizeName	string	see Media Size Self-Describing Names section 5 in [PWG5101.1]	prtOutputDimUnit; prtOutputMaxDimFeedDir prtOutputMaxDimXFeedDir
OutputTrayMinMediaSizeName	string		
Any	various	Extension point to OutputTrayDescription	
OutputTrayStatus	complex		
SubunitStatus	complex	See 3.1.1	
OutputTraySecurity	keyword	*SubunitPresentOnOffWKV*	prtOutputSecurity (PresentOnOff)
OutputTrayRemainingCapacityBasis	keyword	*ObjectCounterBasisWKV*	prtOutputRemainingCapacity
OutputTrayRemainingCapacity	counter	in units specified by OutputTrayCapacityUnit)	prtOutputRemainingCapacity
OutputTrayStackingOrder	keyword	first to last or last to first. See PrtOutputStackingOrderTC in [RFC3805] *OutputTrayStackingOrderWKV;*	prtOutputStackingOrder
OutputTrayPageDeliveryOrientation	keyword	face up or face down See PrtOutputPageDeliveryOrientationTC in [RFC3805] *OutputTrayPageDeliveryOrientationWKV*	prtOutputPageDeliveryOrientation
OutputTrayBursting	keyword	see PresentOnOff in [RFC3805] *SubunitPresentOnOffWKV*;	prtOutputBursting (PresentOnOff)
OutputTrayDecollating	keyword		prtOutputDecollating (PresentOnOff)
OutputTrayPageCollated	keyword		prtOutputPageCollated (PresentOnOff)
OutputTrayOffsetStacking	keyword		prtOutputOffsetStacking (PresentOnOff)
Any	various	Extension point to OutputTrayStatus	
Any	various	Extension point to OutputTray	

3.14 Processors

This complex Element represents the computing and logical Elements that, in conjunction with operational memory, execute the firmware and/or software providing the device functions and features. The Processor corresponds to the System Controller as defined in paragraph 2.2.8 of RFC3805 [RFC3805] It is modeled in Figure 51 with constituent Elements described in Table 35.

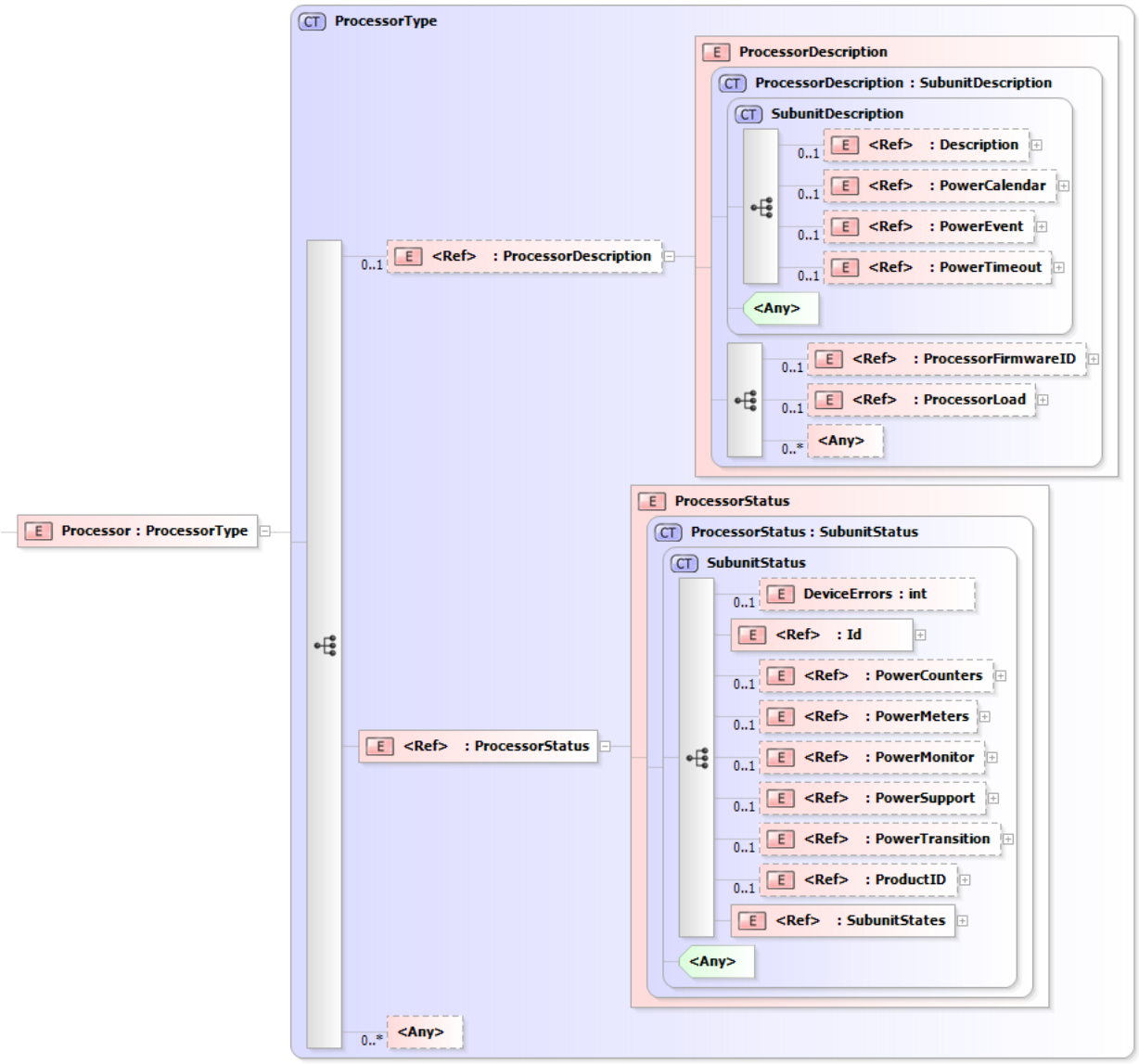

Figure 51 Processor

Table 35 Processor Subunit Elements

Element	DataType	Description	Reference (all [RFC2790])
ProcessorDescription	complex		
SubunitDescription	complex	See 5.1.2	
ProcessorFirmwareID	int	product ID of the firmware associated with the processor identifying manufacturer, model, and version	hrProcessorFrwID
ProcessorLoad	int	percentage of the last minute that processor was not idle	hrProcessorLoad
Any	various	Extension point to ProcessorDescription	
ProcessorStatus	complex	Common Subunit Status	
SubunitStatus	complex	See 5.1.1	prtChannelStatus
Any	various	Extension point to Processor	

3.15 Scan Media Paths

This complex Element represents the media handling aspects of a Scanner Device. It is distinct from the Media Path Subunit (3.11), which refers to the media handling aspects of a Printer Device. Scan Media Paths can contain more than one Scan Media Path Subunit; for example, the platen of a flatbed scanner can be treated as a separate Scan Media Path Subunit from the auto Document feeder. The constituents of the Scan Media Path Subunit complex Element are represented in Figure 52. The Scan Media Path Elements are described in Table 36.

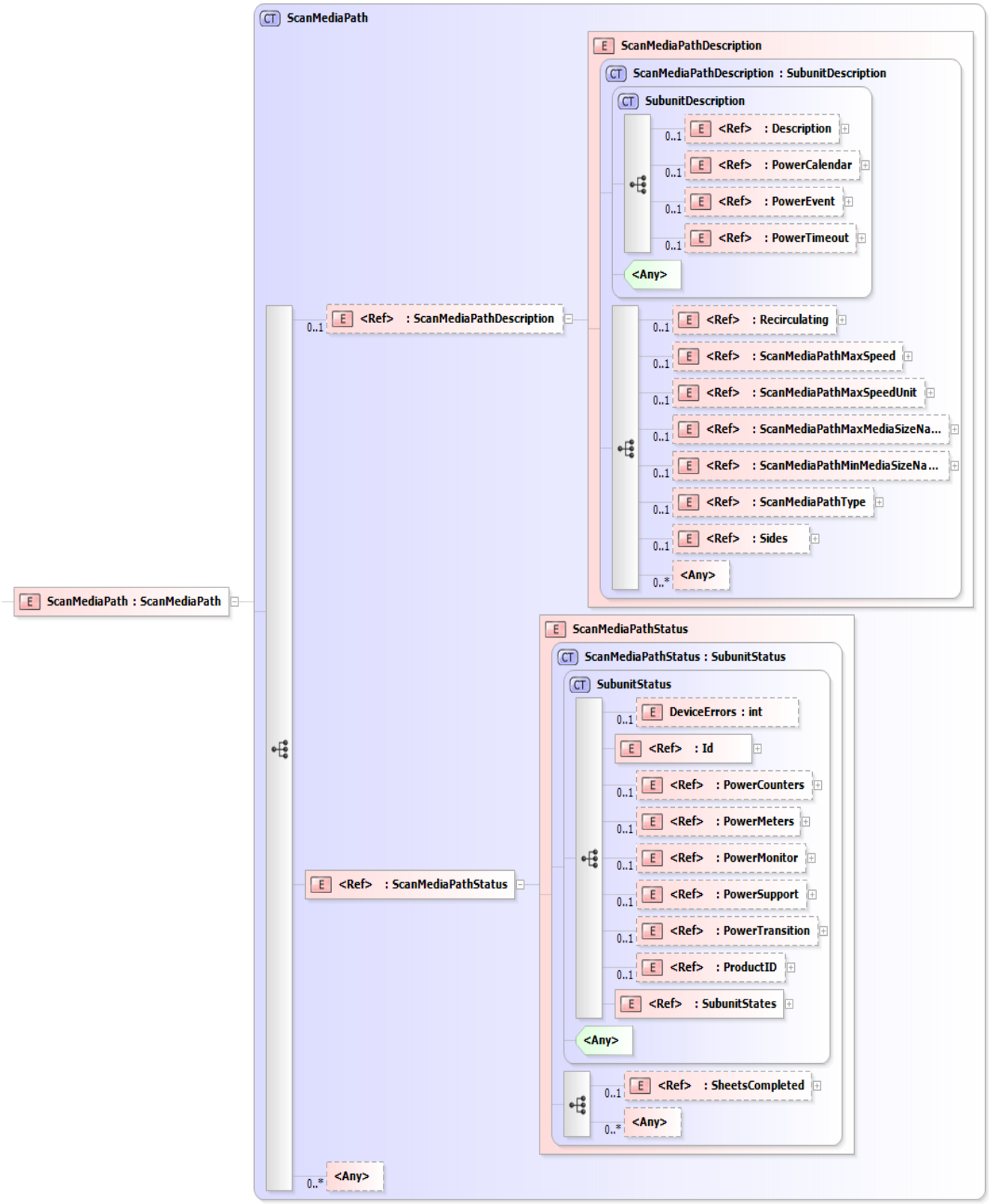

Figure 52 Scan Media Path

Table 36 Scan Media Path Elements

Element	Datatype	Description or *Keyword Group*	Reference
ScanMediaPathDescription	complex		
SubunitDescription	complex	See 3.1.2	
Recirculating	boolean	applicable to Scan Media Paths that are capable of feeding Hard Copy Documents. If value of this Element is 'true' then the Scan Media Path is both an input and output for the Hard Copy Documents. If the Element is missing or the value is set to 'false' then the Hard Copy Document is delivered to an Output Bin after scanning is complete.	[PWG5108.2]
ScanMediaPathMaxSpeed	int		
ScanMediaPathMaxSpeedUnit	keyword	units used for the maximum speed of Scan Media Path. *MediaPathMaxSpeedUnitWKV*	[RFC3805] prtMediaPathMaxSpeedPrintUnit
ScanMediaPathMaxMediaSizeName	keyword	maximum size of media that can flow through the Scan Media Path in Media Self Describing Name see Media Size Self-Describing Names section 5 in [PWG5101.1]; *MediaSizeNameType*	. [PWG5101.1] prtScanMediaPathMediaSizeUnit prtScanMediaPathMaxMediaFeedDir prtScanMediaPathMaxMediaXFeedDir in [RFC3805]
ScanMediaPathMinMediaSizeName	keyword	Minimum size of media that can flow through the Scan Media Path in Media Self Describing Name see Media Size Self-Describing Names section 5 in [PWG5101.1]; *MediaSizeNameType*	. [PWG5101.1] prtScanMediaPathMediaSizeUnit prtScanMediaPathMinMediaFeedDir prtScanMediaPathMinMediaXFeedDir in [RFC3805]
ScanMediaPathType	keyword	type of Scan Media Path. ; *ScanMediaPathTypeWKV*	[RFC3805] prtMediaPathType
Sides	keyword	the ability of the ScanMediaPath to scan multiple sides of an input media sheet. *SidesWKV*	[RFC2911] sides
Any	various	Extension point for ScanMediaPathDescription	
ScanMediaPathStatus	complex		[PWG5108.2]
SubunitStatus	complex	See 5.1.1	
SheetsCompleted	int	number of times a media sheets has passed through the ScanMediaPath.	[PWG5108.2]
Any	various	Extension point for ScanMediaPathStatus	
Any	various	Extension point for ScanMediaPath	

3.16 Scanners

This complex Element represents the image acquisition aspects of a Scanner Device. The constituents of a Scanner Subunit are represented in Figure 53 The constituent Elements are defined in Table 37. The constituent complex Elements Scanner Addressability and Scanner Margin are represented in Figure 54 and Figure 55, with the Elements within these complex Elements identified in Table 38 and Table 39.

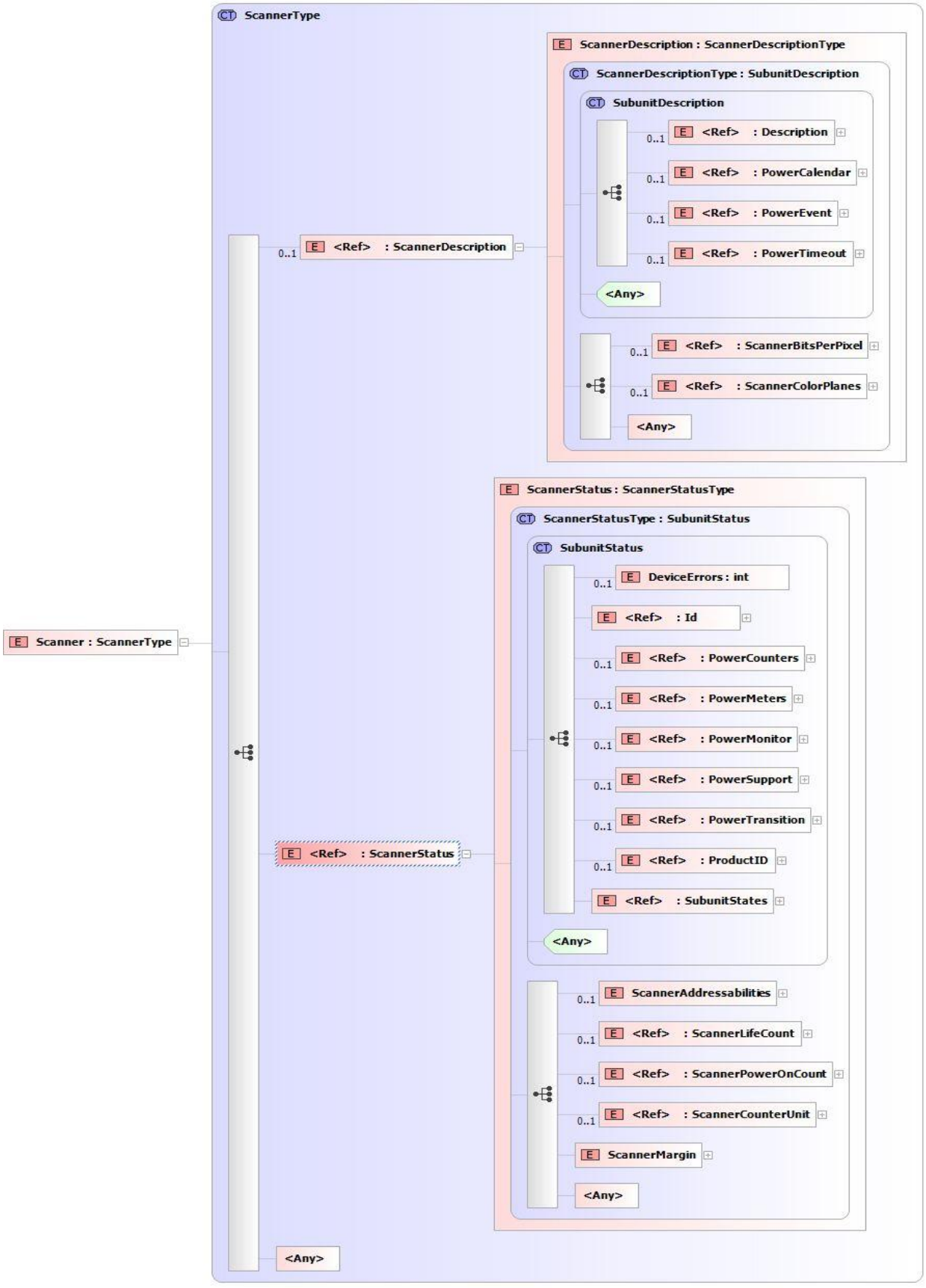

Table 37 Scanner Subunit Elements

Element	Datatype	Description or *Keyword Group*	Reference [RFC3805] *
ScannerDescription	complex		[PWG5108.2]
SubunitDescription	complex	See 3.1.2	
ScannerBitsPerPixel	int	number of bits per pixel used by the scanner's image acquisition subsystem.	[PWG5108.2]
ScannerColorPlanes	int	number of color planes used by the scanner's image acquisition subsystem.	[PWG5108.2]
Any	various	Extension point for ScannerDescription.	
ScannerStatus	complex		[PWG5108.2]
SubunitStatus	complex	See 3.1.1	
ScannerAddressabilities	complex	information on the available addressabilities i.e., resolution of the Scanning hardware. See Table 38	prtMarkerAddressability
ScannerLifeCount	int	number of scans performed by the scanning hardware for the life of the device. The units are measured in ScannerCounterUnits.	prtMarkerLifeCount
ScannerPowerOnCount	int	number of scans performed by the scanning hardware since the device was lasted turned on. The count is in ScannerCounterUnits.	prtMarkerPowerOnCount
ScannerCounterUnit	keyword	units for ScannerCounterLife and ScannerCounterPowerOn . *SubunitCounterUnitWKV*	prtMarkerCounterUnit objects
ScannerMargin	complex	information on the Margin of the Scanning hardware. The margins provide a bounding box within which the Scanner hardware is capable of scanning. See Table 39.	prtMarkerMargin objects
Any	various	Extension point for ScannerStatus	
Any	various	Extenstion point for Scanner Subunit	

*unless otherwise indicated

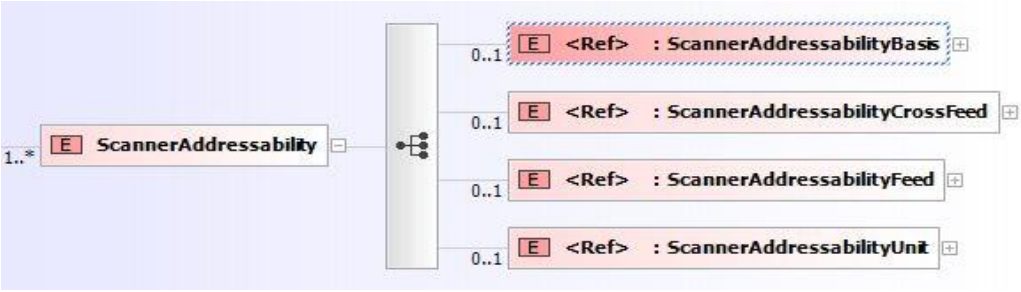

Figure 54 Scanner Addressability

Table 38 Scanner Addressability Elements

Element	Datatype	Description or *Keyword Group*	Reference all [RFC3805]*
ScannerAddressabilityBasis	keyword	confidence level of the addressability information for the Scanning hardware *ObjectCounterBasisWKV*	prtMarkerFeed/XFeedAddressibility out of band values
ScannerAddressabilityCross Feed	int	resolution in the cross feed direction i.e., fast scan direction for the Scanning hardware. This is expressed in ScannerAddressabilityUnits.	prtInterpreterXFeedAddressability
ScannerAddressabilityFeed	int	resolution in the feed direction i.e., slow scan direction for the Scanning hardware. This is expressed in ScannerAddressabilityUnits.	prtInterpreterFeedAddressability
ScannerAddressabilityUnit	keyword	*SubunitAddressabilityUnitWKV*	prtMarkerAddressabilityUnit

*unless otherwise indicated

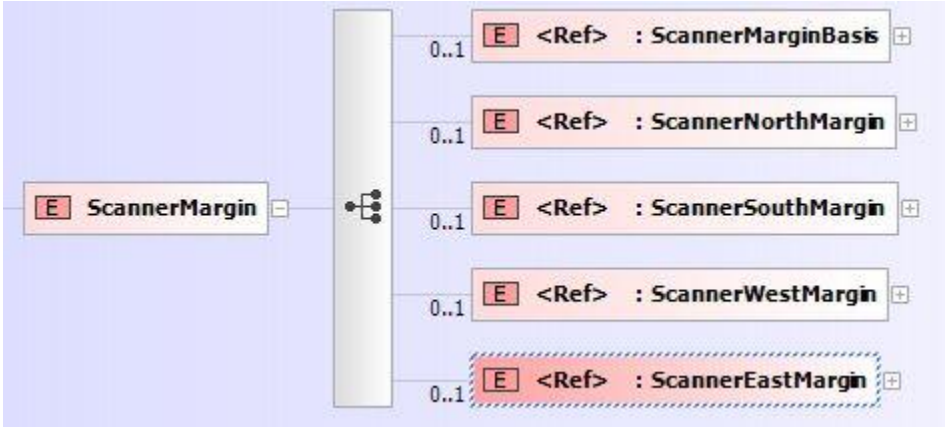

Figure 55 Scanner Margin

Table 39 Scanner Margin Elements

Element	Datatype	Description or *Keyword Group*	Reference all [RFC3805]
ScannerMarginBasis	keyword	confidence level of the Margin information for the Scanning hardware . *ObjectCounterBasisWKV*	prtMarkerNorth/South/East/West out of band values
ScannerNorthMargin	int	offset from the leading edge of the scan hardware or platen where Image acquisition can occur. The units are given in ScannerAddressabilityUnit.	prtMarkerNorthMargin
ScannerSouthMargin	int	offset from the trailing edge of the scan hardware or platen where Image acquisition can occur. The units are given in ScannerAddressabilityUnit.	prtMarkerSouthMargin
ScannerWestMargin	int	offset from the west edge of the scan hardware or platen where Image acquisition can occur. The units are given in ScannerAddressabilityUnit.	prtMarkerWestMargin
ScannerEastMargin	int	This Element expresses the offset from the east edge of the scan hardware or platen where Image acquisition can occur. The units are given in ScannerAddressabilityUnit.	prtMarkerEastMargin

3.17 Storages

This is a sequence of digital data storage Elements, each modeling a storage Subunit used by a Service that provides for digital data storage and recovery as a primary function. The constituents of the Storage Subunit complex Element are represented in Figure 56. A Storage complex Element contains the Elements described in Table 40.

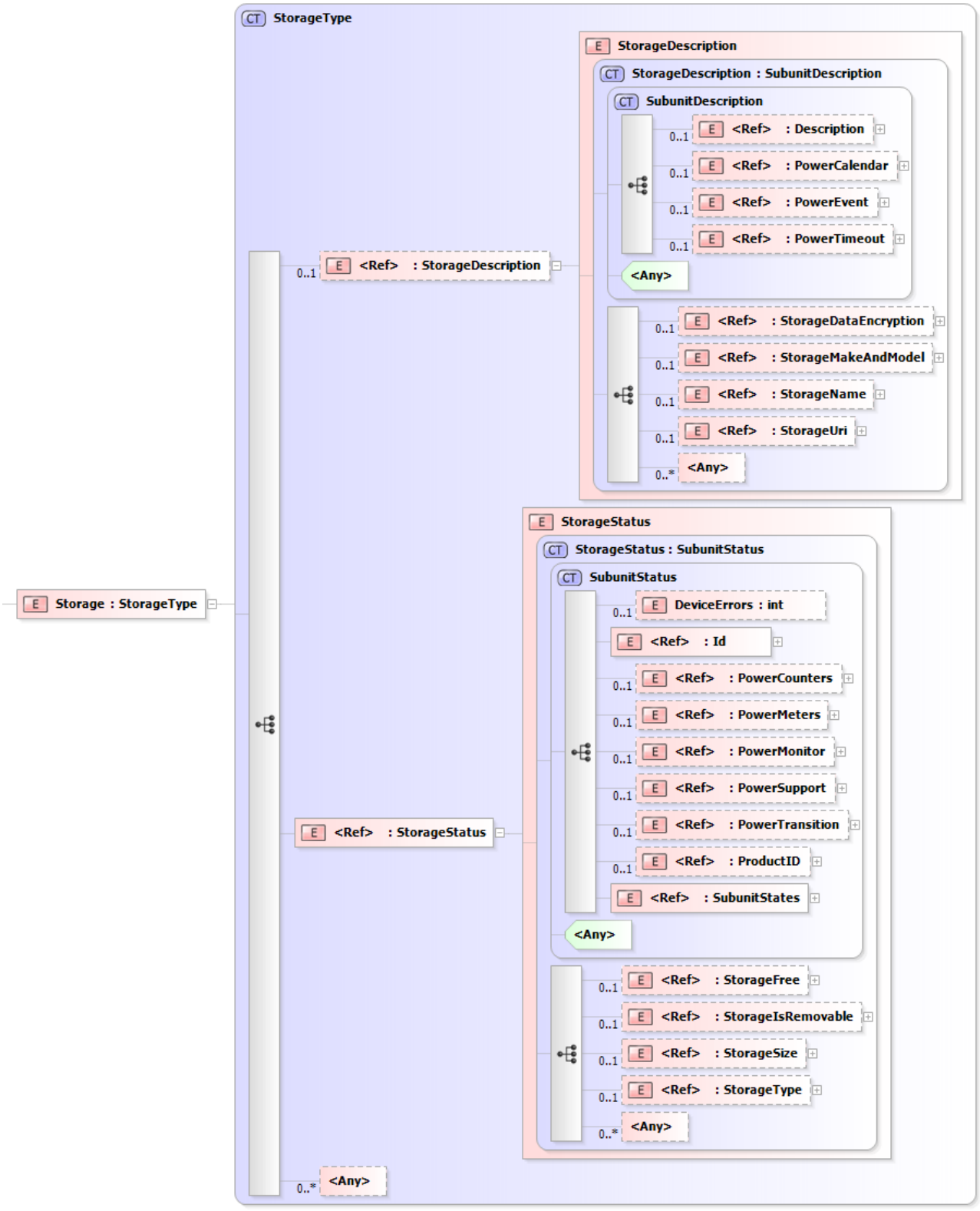

Figure 56 Storage

Table 40 Storage Elements

Element	DataType	Description or *Keyword Group*	Reference
StorageDescription	complex		
SubunitDescription	complex	See 3.1.2	
StorageDataEncryption	boolean	Data encryption is supported	
StorageMakeAndModel	string	information on the manufacturer of the storage Subunit including model name of the storage Subunit	[RFC 2790]
StorageName	string	name of the storage Subunit that is unique within a Service instance.	Values derived from IPPv1.1 ServiceStatus
StorageUri	uri	network location information of the storage Subunit. (applicable only to a NetworkStorage device)	
Any	various	Extension point for StorageDescription	
StorageStatus	complex		
SubunitStatus	complex	See 3.1.1	
StorageFree	int	total bytes currently free for use by the service	hrStorageSize minus hrStorageUsed [RFC 2790]
StorageIsRemovable	boolean	Storage Subunit is removable by the design of its manufacturer See Note.	hrDiskStorageRemoveble [RFC 2790]
StorageSize	int	total number of bytes allocated to the service	hrStorageSize [RFC 2790]
StorageType	keyword	type of the storage Subunit (expanded from HR MIB.) *StorageTypeWKV*	hrStorageType [RFC 2790]
Any	various	Extension point for StorageStatus	
Any	various	Extension point for Storage	

Note: Although the "removable" reference in the HR MIB defines the Boolean in terms of disk storage, in the MFD context it can refer to any type of storage, including flash memory.

3.18 Vendor Subunits

This complex Element allows the vendor to add one or more additional Subunits to his product model. These would be major hardware or software components that do not fit within the set of Subunits listed here, but which act to enhance or give additional functionality to a MFD product. Examples are a thumbprint or access card reader for security. It is modeled in Figure 57 and described in Table 41.

Table 41 Vendor Subunit Elements

Element	Datatype	Description or *Keyword Group*
VendorSubunitDescription	complex	
SubunitDescription	complex	See 3.1.2
Any	various	Extension point for VendorSubunitDescription
VendorSubunitStatus	complex	
SubunitStatus	complex	See 3.1.1
SubunitCounterUnit	keyword	*SubunitCounterUnitWKV*
SubunitLifeCount	counter	
SubunitPowerOnCount	counter	
Any	various	Extension point for VendorSubunitStatus
Any	various	Extension point for VendorSubunit

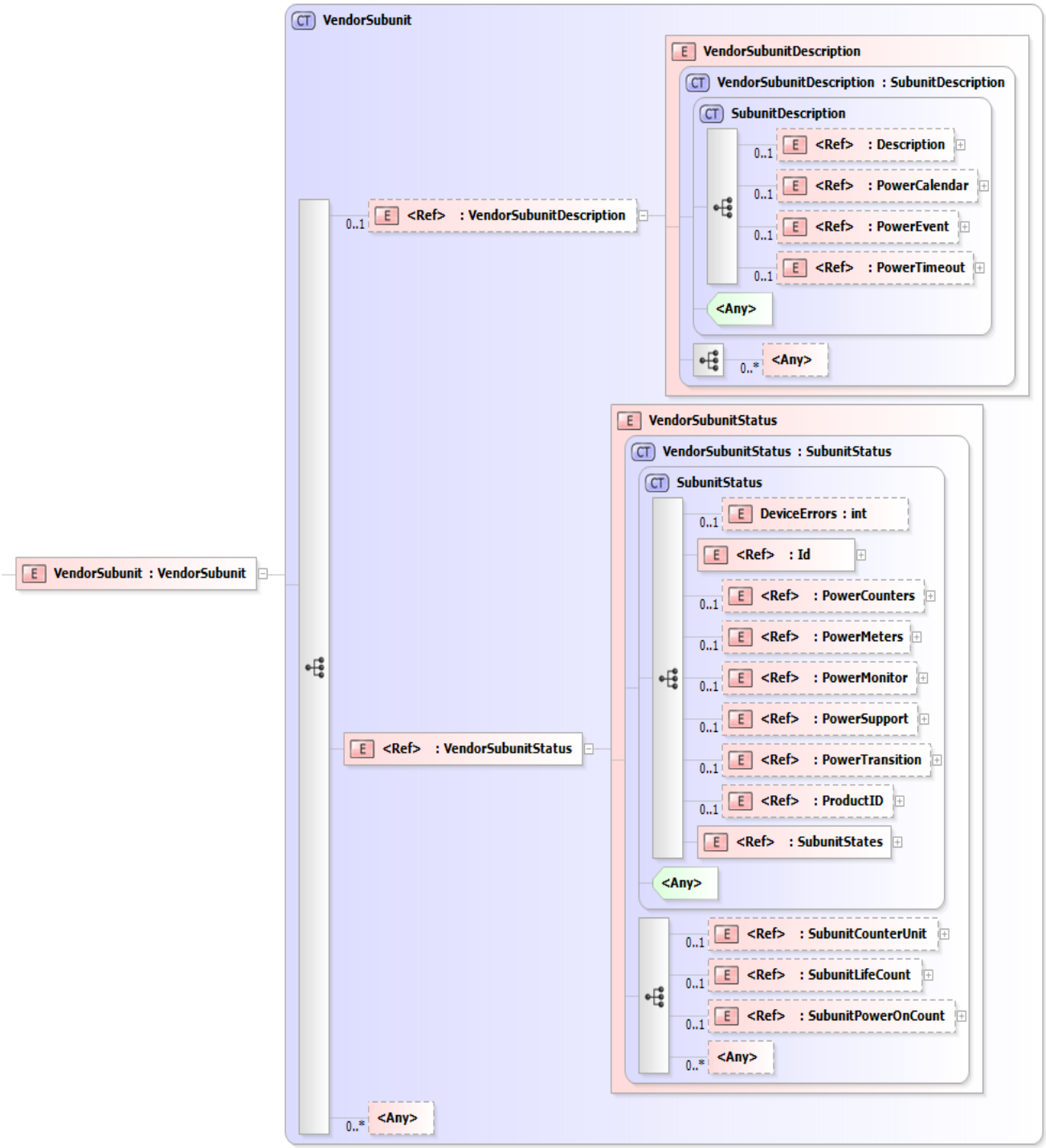

Figure 57 Vendor Subunit

4 Service Model Element Groups

Although the different MFD Services have distinct detailed models, which are described in the Service-specific specifications, the models do have a common basic structure. Further, the unifying concepts described in Chapter 2 and the similarity of aspects of each Service produce some Element commonality. Each MFD Service, and the complex Elements included in it, are composed of Elements which are:

- common to all services (e.g., Service Description, Service Status)
- common to a subset of Services (e.g., Document Processing Capabilities, Job Description Capabilities) and
- unique to a given Service.

This Chapter presents the general model structure of a Service and describes the Elements which are common to all, or to several Services, thereby eliminating the need to repeat these descriptions in each Service specification.

> Note that descriptions and schema diagrams in this Chapter refer to the abstract class “Imaging Service”. This class is included in the PWG MFD XML Schema [MFD_SCHEMA] to assist in understanding the model, but it is not a basic complex Element that is actually contained in any real MFD Service. Rather, the schema representation of each individual MFD Service includes Elements corresponding to those in the abstract Imaging Service that are applicable to that individual MFD Service. The names of Elements in the abstract class are prefixed with string ”Imaging” while the names of corresponding Elements in the actual MFD Service are prefixed with the Service name (e.g., “Print”, “Scan”.) For example, “Imaging Service Capabilities” in the abstract class corresponds to “Copy Service Capabilities” in the Copy Service. The following descriptions should be read with the understanding that not all of the Elements described in the abstract Imaging Service are necessarily included in each actual MFD Service.

The abstract Imaging Service is represented in Figure 58. It is composed of a Job Table and the Element groups in the Imaging Service type:

- Available Job Ticket
- Service Capabilities
- Service Capabilities Ready
- Service Configuration-
- Service Defaults
- Service Description
- Service Status
- Any (for PWG or vendor extension)

Services that do not process Jobs (e.g., Resource Service) do not include a Job Table or Service Defaults. Those services do not process Jobs nor act on tickets; rather, they simply process requests. Services to which a user typically supplies a Job Ticket do not include an Available FaxIn Job Ticket. That is, the Available Job Ticket is included only for those Services that have jobs created by inbound traffic (e.g., FaxIn, EmailIn).

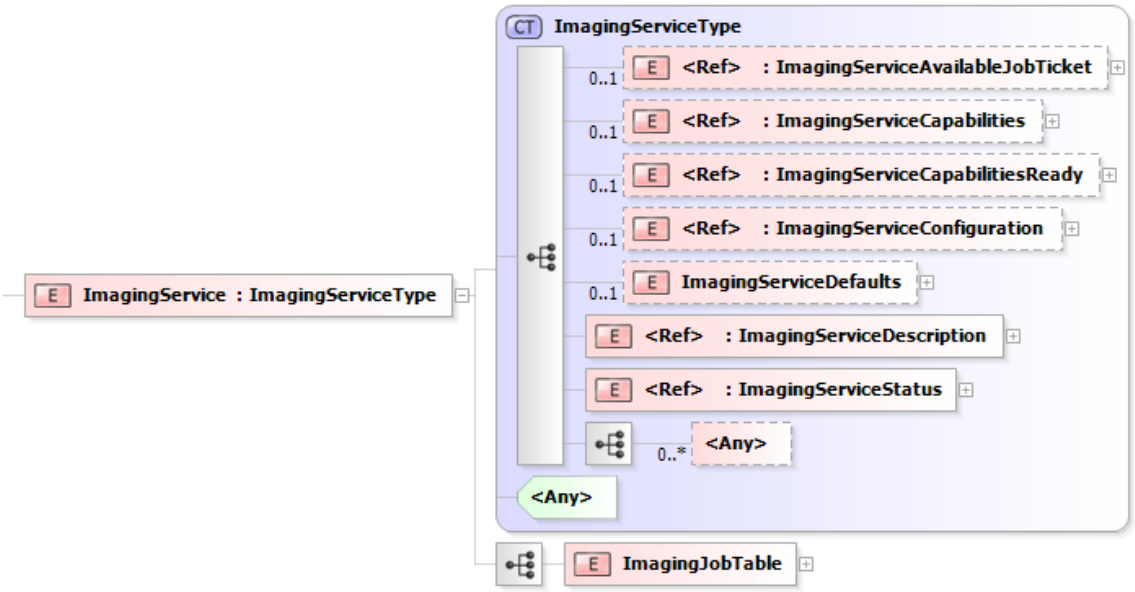

Figure 58 Abstract Imaging Service Structure

4.1 Job Table

Each Job-handling Service includes a Job Table that is constituted of two lists, as represented in Figure 59:

- Pending and Active Jobs appear in the Active Jobs list.
- Jobs that have reached a terminal state (i.e., Completed, Aborted, and Canceled) appear in the Job History list.

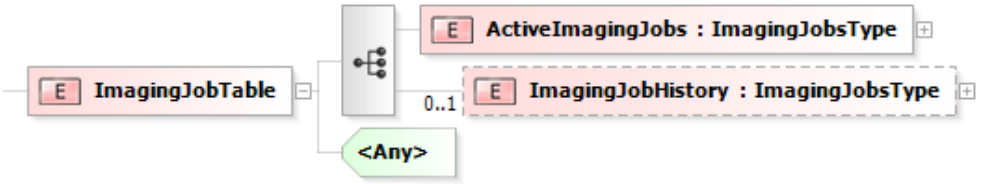

Figure 59 Imaging Job Table

The Job History is optional and the amount of time a Job is retained in the Job History list is implementation specific.

The same Jobs model is used in Active Jobs and Job History and is represented in Figure 60.

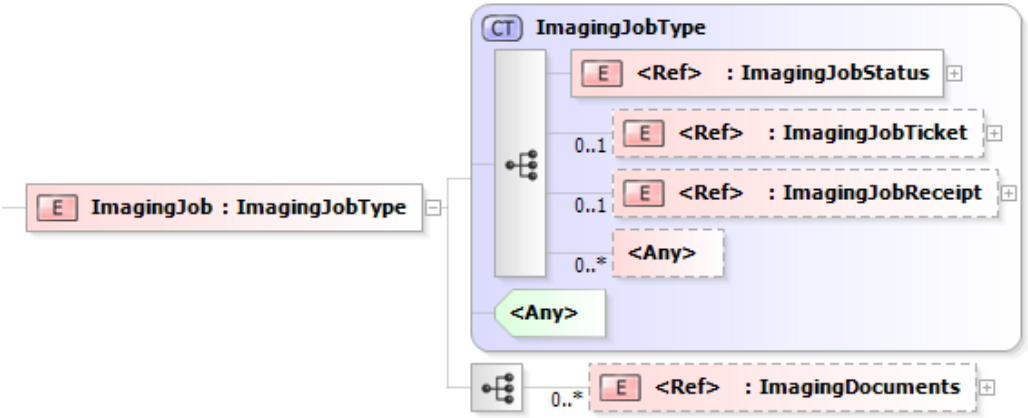

Figure 60 Imaging Job Model

The Job Element is considered as including both overall Job Elements (Job Status, Job Ticket and Job Receipt, of which only Job Status is mandatory); and optionally, Documents, consisting of zero or more Document Elements. The Document is itself a complex Element.

Because the Job Element and the Document Element are each fairly complicated, their models are separately discussed in Chapters 5 and 6 respectively.

4.2 Service Available Job Ticket

MFD Services that receive an implied Job Create request by an incoming FaxIn or EmailIn transmission typically do not received a Job Ticket with the request. However, there may be information in or associated with the transmission that contain implicit Job Ticket information, such as destination or origination address, or time of day. Therefore, a single Default Job Ticket would not be adequate. Such Services may therefore include a Service Available Job Ticket, formed as shown in Figure 61.

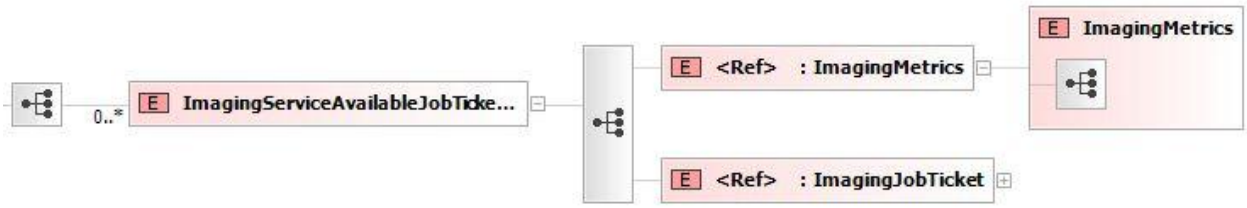

Figure 61 Imaging Service Available Job Ticket

The Available Job Ticket is composed of a basic ImagingJobTicket complex Element, itself of a form common to all Job Tickets and described in section 5.2, and an ImagingMetrics Element. This ImagingMetrics Element provides for the conditions associated with or contained with the transmission that may impact the values of Elements in the Job Ticket. The form and contents of ImagingMetrics are Service dependent, and are discussed in the specifications of the appropriate Services.

4.3 Service Capabilities and Service Capabilities Ready

The Service Capabilities provide information about the Elements that can be used in Job Tickets and Document Tickets to describe the desired processing of a Job and its Documents. Some Services also include a Service Capabilities Ready Element, of the same form and content as Service Capabilities. The Service Capabilities Ready Element provides information about the Elements that can be used in Job Tickets and Document Tickets without operator intervention. For example, a printer that can accept media of a given size would have that media size listed under Print Service Capabilities Media Size Name. However, if use of that media size required a particular input tray that was not mounted, that media size would not be listed under Print Service Capabilities Ready Media Size Name.

The Imaging Service Capabilities (Figure 62)) and Imaging Service Capabilities Ready Elements are grouped into:

- Document Ticket Capabilities
    - Document Processing Capabilities:
    - Document Description Capabilities
- Job Ticket Capabilities
    - Document Processing Capabilities:
    - Job Description Capabilities
    - Job Processing Capabilities

These names reflect that fact that Service Capabilities are the processing and description elements that can be requested (and presumably satisfied) in the Job and Document Tickets. Because Document Processing instructions can be in either Document or Job Tickets, identical Document Processing Capabilities structures are included for both Document and Job Ticket Capabilities.

> Note that the “Capabilities” Elements identified in this Chapter have the same names as the Elements that may be included in the Job or Document Ticket, identified in Chapter 5 and Chapter 6, respectively. However, a “capabilities” Element is typically multivalued because it must specify all allowable values, and this is reflected its syntax. The corresponding Job Ticket or Document Ticket Element has the specific value to be used for the Job. For example, the Input Source Capabilities Element is a sequence of keywords while the Input Source Ticket Element is a single keyword.

The following sections tabulate the Elements in each of the Service Capabilities Element groups, with the data type and brief description of each Element. Where the data type is keyword or list of keywords, the table identifies the keyword group. In general, the keywords will be either designated WKV (Well Known Value) or Extension Patterns WKV keyword groups contain “fixed” values defined by specification; please refer to the keyword group in the PWG MFD XML Schema [MFD_SCHEMA] for an explicit list of the applicable keyword values. Keyword groups that include Extension Patterns refer to a vendor’s extension keyword group. These are typically unions with a defined pattern assigned by vendors to extend the keyword value set.

The capabilities typically include both Elements inherited from the imaging service class and service dependent Elements, although there may be some commonality of these latter Elements among Services.

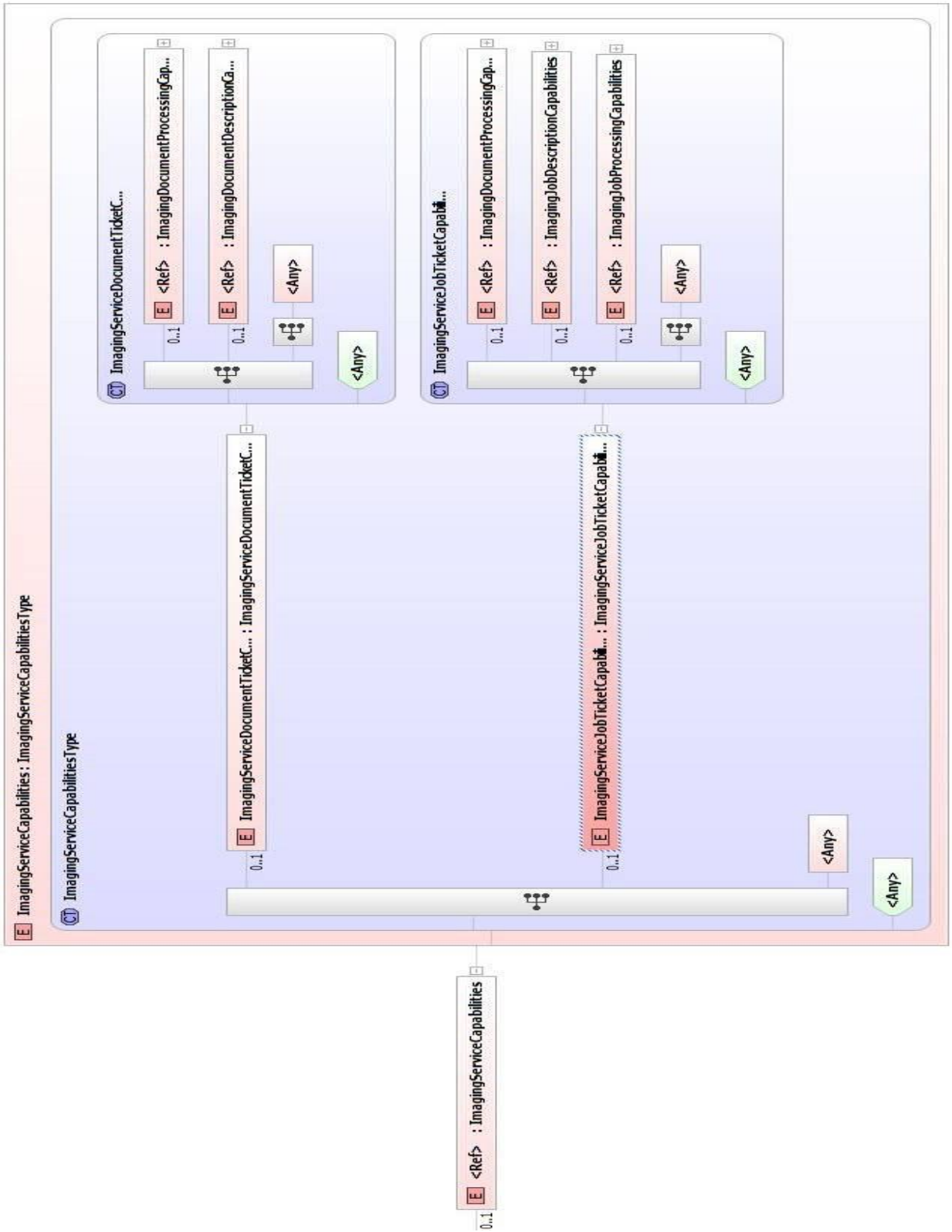

Figure 62 Imaging Service Capabilities

4.3.1 Service Job and Document Ticket Document Processing Capabilities

Document Processing Capabilities are System Capabilities that identify the information about the Document Processing Elements that can be used in Job Tickets and Document Tickets. Document Processing Capabilities consists of two sequences:

- Elements inherited from the Service super class as modeled in Figure 63, including Presentation Direction Number Up; Number Up. These elements are listed in Table 42.
- Service specific Elements (a much larger set) that may be common to two or more Services. Generally, these elements are common among image-outputting Services such as Scan, FaxOut and EmailOut (Figure 64 and Table 43) , and among impression-outputting Services such as Print, FaxIn and EmailIn (Figure 65, Figure 66 and Figure 67 and Table 44).

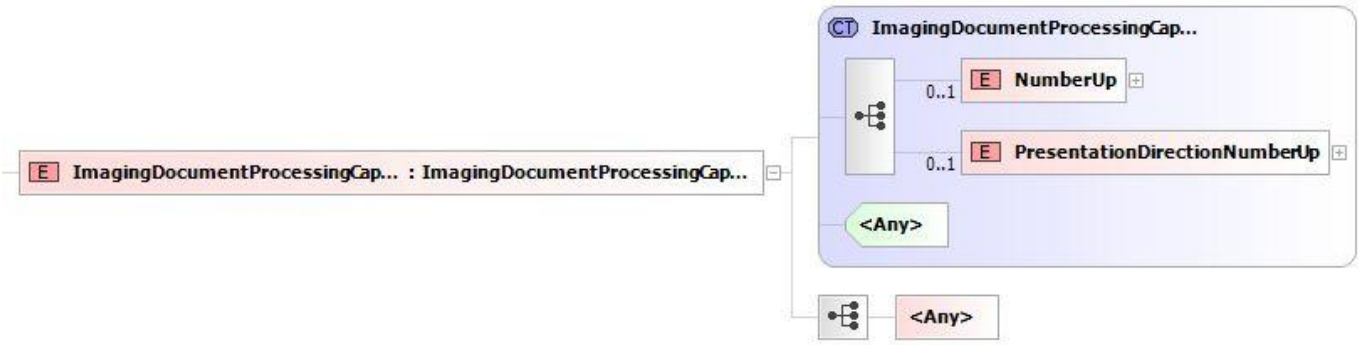

Figure 63 Imaging Document Processing Capabilities

Table 42 Imaging Document Processing Capabilities

Element	DataType	Description or *Keyword Group*
NumberUp	list of int	list of number of input “pages” that are included in a single output “impression; or the number of input “images” that are included in an output image.
PresentationDirectionNumberUp	list of keywords	supported placement order of the input pages or images in the finished output. *PresentationDirectionNumberUpWKV*
Any	various	Extension point for ImagingDocumentProcessingCapabilities

The Service-specific Document Processing Capability elements for the Scan Service are representative of the Service-specific Document Processing Capability elements for image-outputting Services such as EmailOut and FaxOut. The elements are shown schematically in Figure 64 and identified in Table 43.

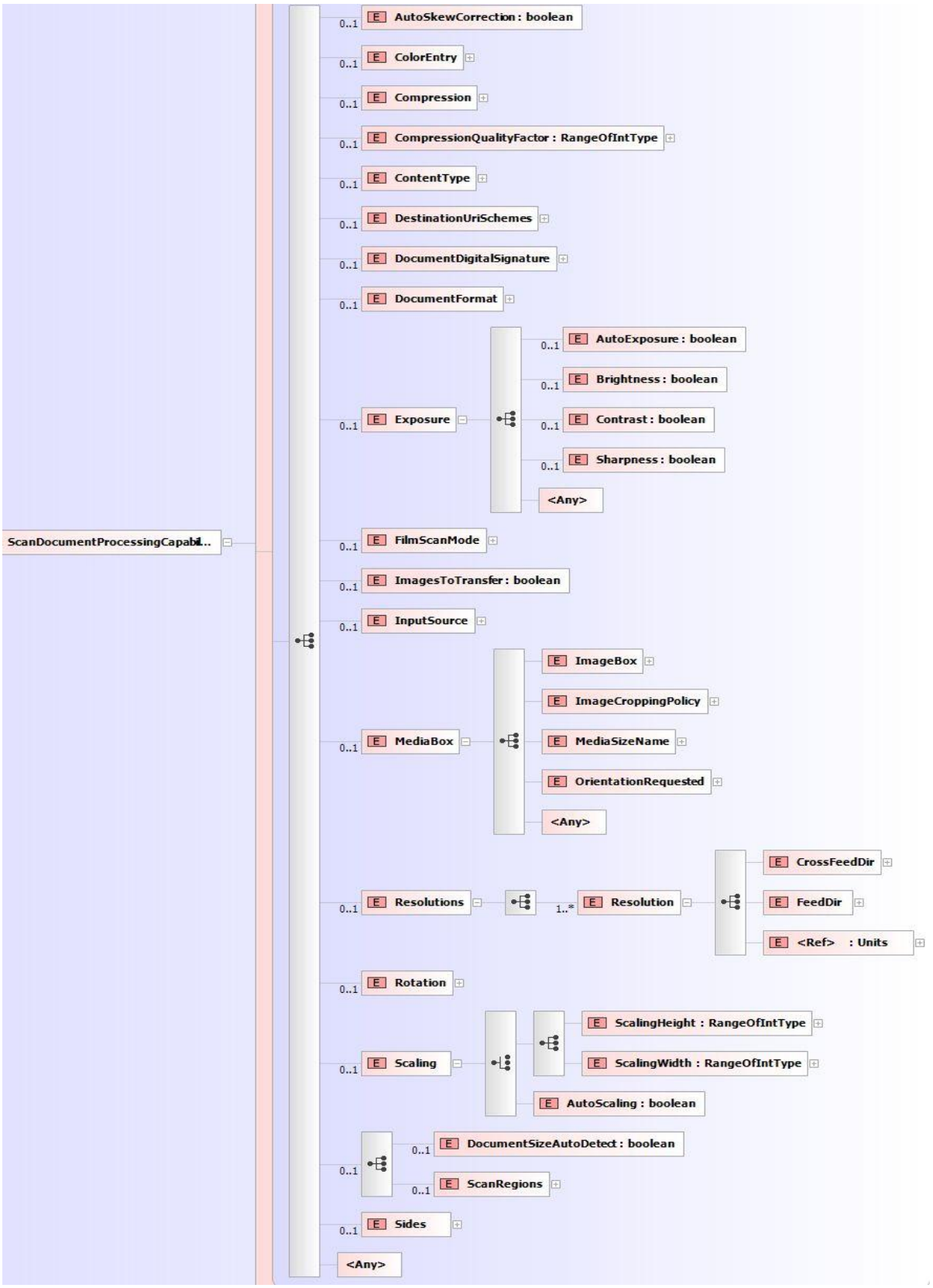

Figure 64 Representative Service-Specific Document Processing Capabilities (Scan Service)

Table 43 Service-Specific Document Processing Capabilities Image Output

Element	DataType	Description or *Keyword Group*
AutoSkewCorrection	boolean	Service's ability to detect and correct small skew orientation errors resulting from placement of the original Document relative to the Scan Subunit
ColorEntry	list of keywords	color related characteristics for the output; e.g., BlackAndWhite1, Grayscale4, RGB24, CMYK32. *ColorEntryWKV, KeywordNsExtensionPattern*
Compression	list of keywords	the compression algorithms supported by the Service. *CompressionWKV*
CompressionQualityFactor	range of int	range of compression factors supported
ContentType	list of keywords	main characteristics of a Document as supported by the Service. *ContentTypeWKV*
DestinationUriSchemes	list of strings	URI Schemes supported that can be used to transfer a Digital Document to its Destination. See [RFC4395]
DocumentDigitalSignature	list of keywords	types of DigitalDocument digital signatures supported See [PWG5100.7] paragraph 3.2.3 [PWG5100.5] paragraph 9.1.11 *DocumentDigitalSignatureWKV, KeywordNsExtensionPattern*
DocumentFormat	list of keywords	Digital Document formats supported by the Service. values are MIME types. *DocumentFormatWKV*
Exposure	complex	Elements relating to the perceived quality of a scanned Image
AutoExposure	boolean	if true, Service supports automatic adjustment of Brightness Contrast and/or Sharpness.
Brightness	boolean	if true, Service supports operator control of brightness i
Contrast	boolean	if true, Service supports operator control of contrast
Sharpness	boolean	if true, Service supports operator control of sharpness
Any	various	Extension point for Exposure
FilmScanMode	list of keywords	supported exposure types for scanning film *FilmScanModeWKV, KeywordNsExtensionPattern*
ImagesToTransfer	boolean	If true, Service supports specifying the number of scanned Images in a output DigitalDocument
InputSource	list of keywords	source of the scanned Document supported. *InputSourceWKV*
MediaBox	complex	
ImageBox	complex	
Height	range of int	set of ranges of integers bound to the RegionUnit
Width	range of int	
X-Offset	range of int	
Y-Offset	range of int	
RegionUnits	keyword	*RegionUnitWKV*
MediaSizeName	list of keywords	MediaSizeNameType
OrientationRequested	list of keywords	OrientationRequestedType
Any	various	Extension point for MediaBox
Resolutions	complex	See [RFC2911]
Resolution	complex	the supported sequence of resolutions in Cross Feed and Feed direction for image capture. [RFC2911 para 4.1.15]
CrossFeedDir	int	allowed values for resolutions, in Units, in CrossFeed direction [RFC2911 para 4.1.15]
FeedDir	int	allowed values for resolutions, in Units, in Feed direction [RFC2911 para 4.1.15]
Units	list of keywords	*UnitsWKV*
Rotation	list of int	the supported rotation in degrees clockwise for a scanned Document. Vendors may extend the allowed values *RotationWKV* (allowed values: 0, 90, 180, 270)

Element	DataType	Description or *Keyword Group*
Scaling	complex	the support for scaling.
ScalingHeight	range of int	supported range of the Scaling in the fast scan direction. A value of '100' specifies that no adjustments are made to the scanned Image. Scaling is expressed in 1 percent increments. Values below '100' reduce the magnification and values above increase magnification.
ScalingWidth	range of int	supported range of the Scaling in the fast scan direction. A value of '100' specifies that no adjustments are made to the scanned Image. Scaling is expressed in 1 percent increments. Values below '100' reduce the magnification and values above increase magnification.
AutoScaling	boolean	the support for automatically adjusting the Scaling of the image from the Hardcopy Document in an implementation specific manner
DocumentSizeAutoDetect	boolean	[WS-Scan] para.4.5.2.2.7.1
Scan Regions	complex	[WS-Scan] para. 4.5.2.2.11.1.1
Scan Region	complex	[WS-Scan] para. 4.5.2.2.11.1.1
Height	range of int	set of ranges of integers bound to the RegionUnit
Width	range of int	
X-Offset	range of int	
Y-Offset	range of int	
ContentRegionUnits	keyword	*RegionUnitWKV*
Sides	list of keywords	*SidesWKV*
Any	various	Extension point for <service> DocumentProcessingCapabilities

The Service-specific Document Processing Capability elements for the Print Service are representative of the Service-specific Document Processing Capability elements for impression-outputting Services such as EmailIn and FaxIn. The elements are shown schematically in Figure 65, Figure 66 and Figure 67 and and are identified in Table 44.

Note that, as a Document Processing Element, MediaCol (Collection of attributes characterizing the media upon which the impressions are made) is a complex element with many members. It appears in many places in the model. However, within capabilities, the Element of this name is a simple Boolean Element, indicating whether or not that MediaCol is supported in conjunction with defining the media to be used for some purpose. Capabilities also includes a MediaCol Supported complex element which includes all of the constituent elements of MediaCol and the values supported by the Service for each of these elements. The description of MediaCol Supported, including the constituent elements, is in 4.3.1.1.

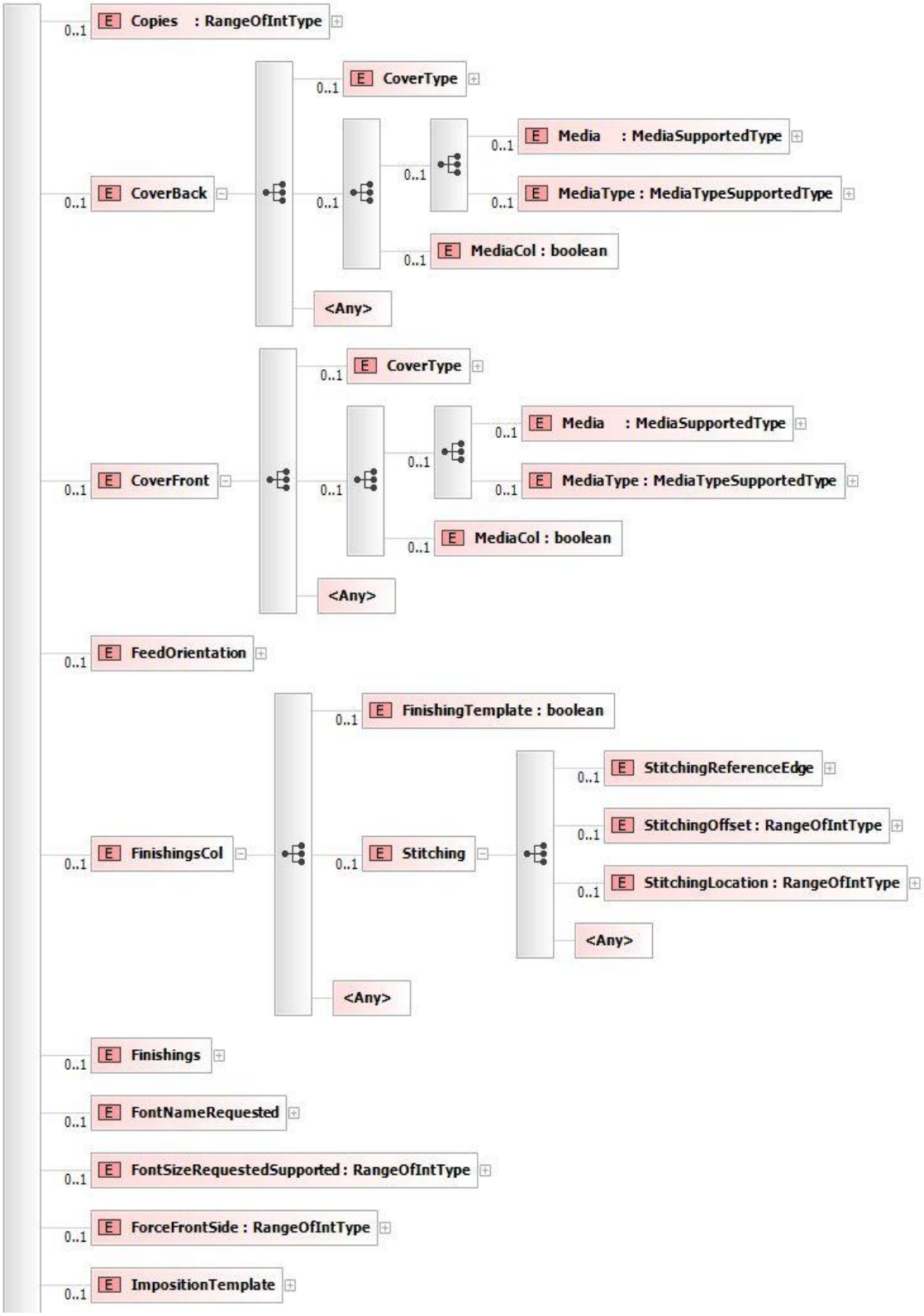

Figure 65 Representative Service-Specific Document Processing Capabilities (Print)-Sheet 1

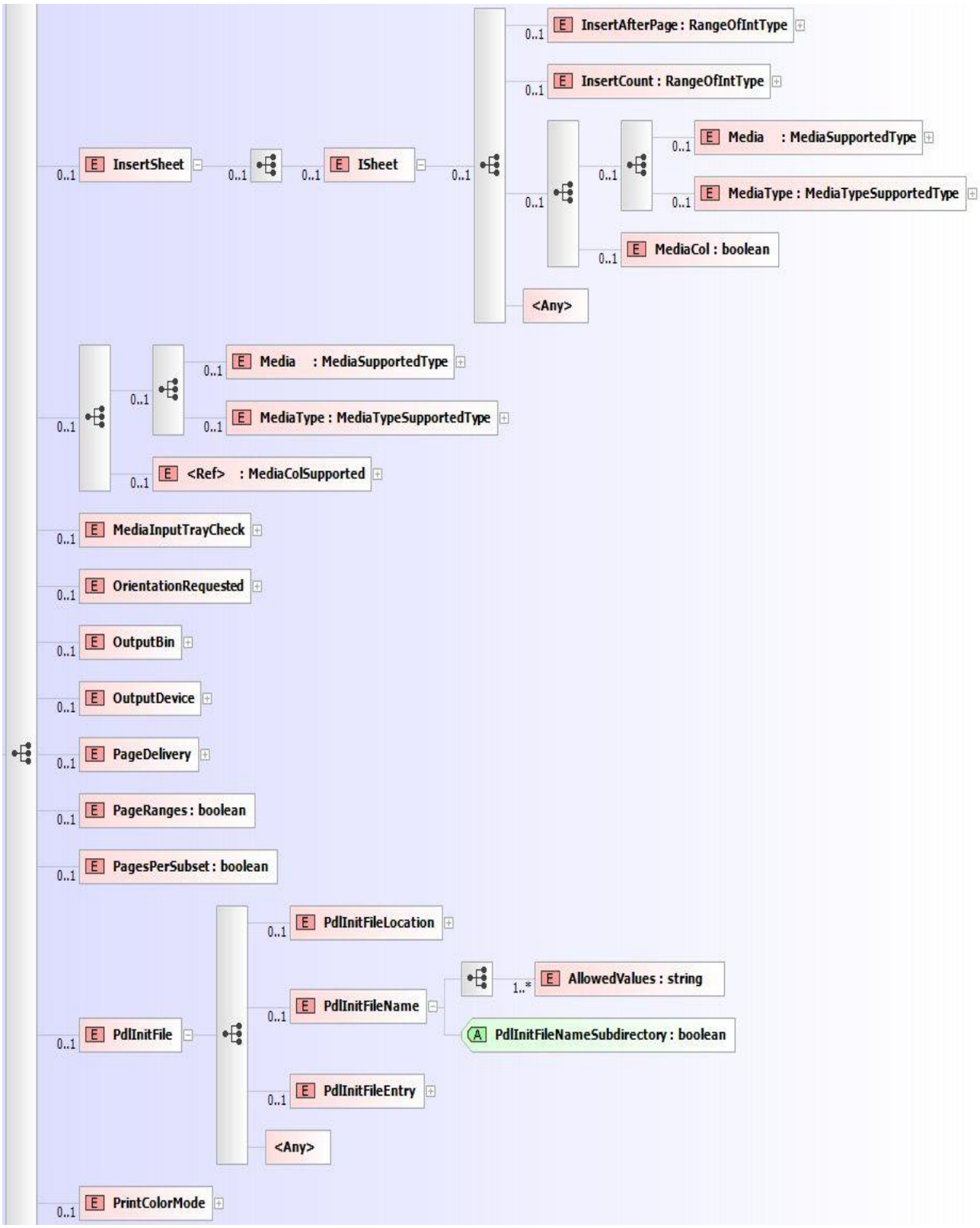

Figure 66 Representative Service-Specific Document Processing Capabilities (Print) Sheet 2

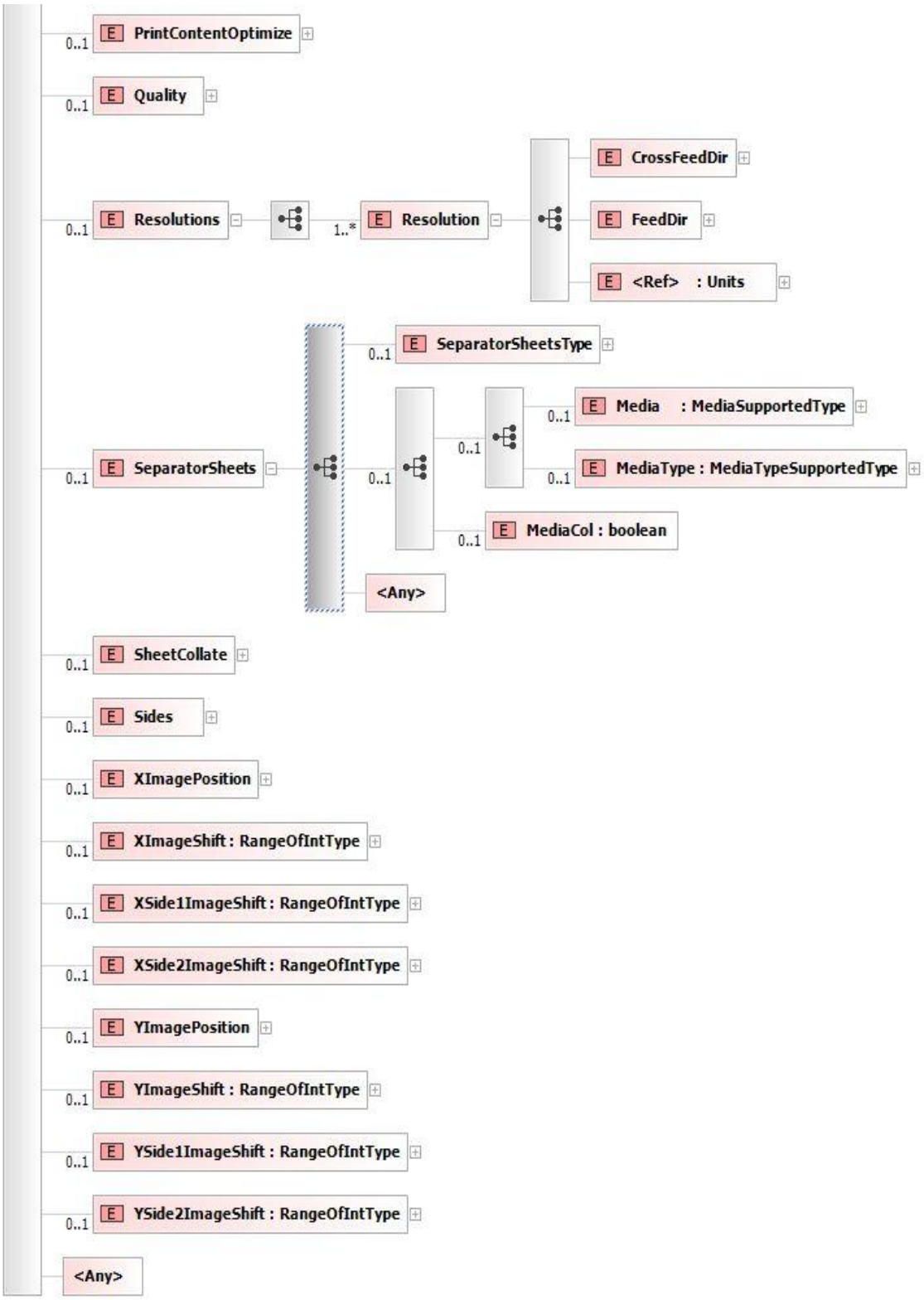

Figure 67 Representative Service-Specific Document Processing Capabilities (Print) Sheet 3

Table 44 Service-Specific Document Processing Capabilities Impression Outputs

Element	DataType	Description or *Keyword Group*
Copies	range of int	
CoverBack	complex	
CoverType	list of keywords	*CoverTypeWKV, KeywordNsExtensionPattern*
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*
MediaType	list of keywords	*MediaTypeWKV*
MediaCol	boolean	Indicates whether feature is supported
Any	various	Extension point for CoverBack
CoverFront	complex	
CoverType	list of keywords	*CoverTypeWKV, KeywordNsExtensionPattern*
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*
MediaType	list of keywords	*MediaTypeWKV*
MediaCol	boolean	Indicates whether feature is supported
Any	various	Extension point for CoverFront
FeedOrientation	list of keywords	*FeedOrientationWKV KeywordNsExtensionPattern*
Finishings	list of keywords	*FinishingsWKV, KeywordNsExtensionPattern*
FinishingsCol	complex	
FinishingTemplate	boolean	
Stitching	complex	
StitchingReferenceEdge	list of keywords	*StitchingReferenceEdgeWKV, KeywordNsExtensionPattern*
StitchingOffeset	range of int	
StitchingLocation	range of int	
Any	various	Extension point for Stiching
Any	various	Extension point for FinishingsCol
FontNameRequested	list of strings	
FontSizeRequestedSupported	range of int	
ForceFrontSide	range of int	
HeaderPrint	list of keywords	HeaderPrintType (e.g., FaxOut and EmailOut)
ImpositionTemplate	list of strings	
InsertSheet	complex	
Isheet		
InsertAfterPage	range of int	
InsertCount	range of int	
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*
MediaType	list of keywords	*MediaTypeWKV*
MediaCol	boolean	Indicates whether feature is supported
Any	various	Extension point for InsertSheet
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*
MediaType	list of keywords	
MediaColSupported	complex	Zero or more instances of MediaCol. See 4.3.1.1
MediaCol	complex	The Media Col complex element appears in several places and is separately described. See Table 45
MediaInput TrayCheck	list of keywords	*MediaInput TrayCheckWKV*, MediaNsExtensionPattern
OrientationRequested	list of keywords	*OrientationRequestedWKV*
OutputBin	list of keywords	*OutputBinWKV, KeywordNsExtensionPattern*
OutputDevice	string	

Element	DataType	Description or *Keyword Group*
PageDelivery	list of keywords	*PageDeliveryWKV*, KeywordNsExtensionPattern
PageRanges	boolean	if true, Service supports this feature
PagesPerSubset	boolean	if true, Service supports this feature
PdlInitFile	complex	(Note: PdlInitFiles is an artifact of the XML encoding)
PdlInitFileLocation	list of strings	specifies the path(s) to the directory (directories) of the supported initialization file(s) [PWG5100.11] paragraph 7.10.1.1.1
PdlInitFileName	list of strings	name(s) of the supported initialization file(s) [PWG5100.11] para 7.10.1.2.1
PdlInitFileNameSubdirectory	boolean (attribute of PdlInitFileName	indicates whether or not the Printer will accept a FORWARD SLASH ("/") character in the value of the "PdlInitFileName" [PWG5100.11] paragraph 7.10.1.2.2
PdlInitFileEntry	list of strings	specifies the name(s) of the supported entry points within the initialization file(s) [PWG5100.11] paragraph 7.10.1.3.1
Any	various	Extension point for PdlInitFile
<service>ColorMode	list of keywords	*PrintColorModeWKV*
<service>tContentOptimize	list of keywords	*PrintContentOptimizeWKV, KeywordNsExtensionPattern*
Quality	list of keywords	*PrintQualityWKV, KeywordNsExtensionPattern*
Resolutions	complex	
Resolution	complex	the supported sequence of resolutions in Cross Feed and Feed direction for image capture.
CrossFeedDir	int	
FeedDir	int	
Units	list of keywords	*UnitsWKV*
SeparatorSheets	complex	
SpreaderSheetsType	list of keywords	*SeparatorSheetsTypeWKV, KeywordNsExtensionPattern*
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*
MediaType	list of keywords	*MediaTypeWKV*
MediaCol	boolean	Indicates whether feature is supported
Any	various	Extension point for SeparatorSheets
SheetCollate	list of keywords	*SheetCollateWKV*
Sides	list of keywords	*SidesWKV*
XImagePosition	list of keywords	See [PWG5100.3] *XImagePositionWKV*
XImageShift	range of int	See [PWG5100.3]]
XSide1ImageShift	range of int	
XSide2ImageShift	range of int	
YImagePosition	list of keywords	See [PWG5100.3]] *YImagePositionWKV*
YImageShift	range of int	See [PWG5100.3]
YSide1ImageShift	range of int	
YSide2ImageShift	range of int	
Any	various	Extension point for <service> DocumentProcessingCapabilities

4.3.1.1 Media Collection Supported

The collection of media attributes, grouped in the Media Col complex element, appears both in Document Processing and Job Processing. Therefore, Media Col Supported appears in both Document Processing Capabilities and Job Processing Capabilities. The Media Col structure is shown in Figure 68 with the constituent capabilities elements listed in Table 45. As with other capabilities Element values, Media Col

Supported elements are largely multivalued. Within an actual Job or Document Ticket, these elements would typically have a single value.

Media Col contains many member Elements. Therefore, unlike for other capabilities Elements, Services should not return Media Col Supported values in response to a general Get<service>Elements request. Rather, an explicit request for the specific Media Col elements desired is required.

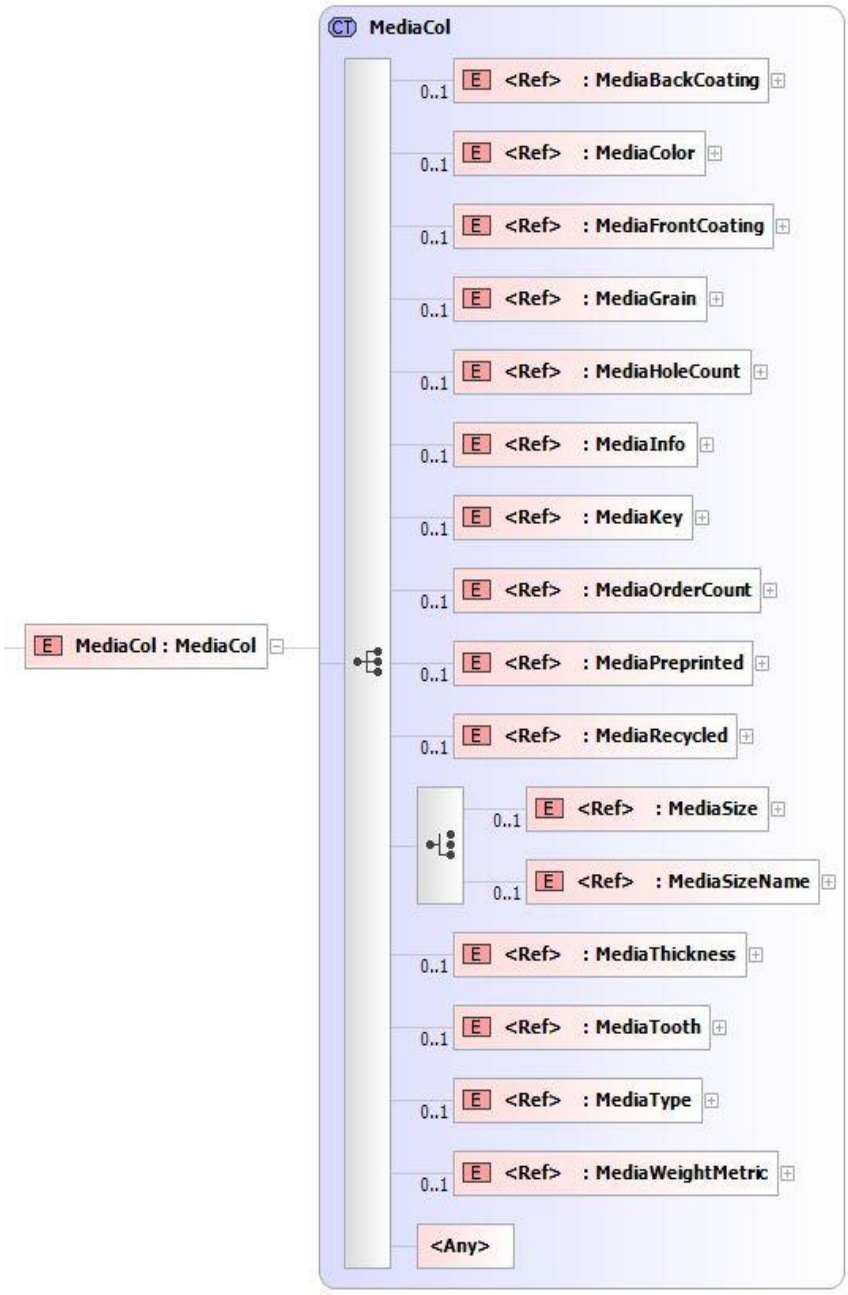

Figure 68 Media Col Structure

Table 45 MediaCol Elements

Element	DataType	Description or *Keyword*	Reference [PWG5100.3]
MediaCol			
MediaBackCoating	list of keywords	supported pre-process coatings applied to the media *MediaCoatingWKV*, MediaColorExtensionPattern	para 3.13.10
MediaColor	list of keywords	media color supported *MediaColorWKV*, MediaColorExtensionPattern	para 3.13.4
MediaFrontCoating	list of keywords	supported pre-process coating applied to the media *MediaCoatingWKV*, MediaNsExtensionPattern	para 3.13.10
MediaGrain	list of keywords	Supported grain of the media (affects curling and folding *MediaGrainWKV*, MediaNsExtensionPattern	[PWG5100.11] para 11.5.2
MediaHoleCount	range of int	Supported number of predrilled holes for the media	para 3.13.6
MediaInfo	boolean	If true, descriptive information for the media is supported	para 3.13.3
MediaKey	boolean	If true, service supports unique name of the media 1	para 3.13.
MediaOrderCount	range of int	number of sheets, within an ordered sequence of sheets; after which the sequence begins to repeat permitted by the service	para 3.13.7
MediaPreprinted	list of keywords	Indicates supported pre-printed characteristics of the media *MediaPreprintedWKV*, MediaNsExtensionPattern	para 3.13.5
MediaRecycled	list of keywords	Indicates the supported recycled characteristics of the media *MediaRecycledWKV*, MediaNsExtensionPattern	para 3.13.11
MediaSize	complex	numerical media width and height dimensions supported by the service	para 3.13.8
XDimension	range of int		para 3.13..8.1
YDimension	range of int		para 3.13..8.2
MediaSizeName	list of keywords	Supported named sizes of the media *MediaSizeNameWKV*, MediaSizeNameExtensionPattern	[PWG5101.1] para 5
MediaThickness	range of int	Supported thickness of the media in hundredths of millimeters]	[PWG5100.11p ara 11.5.3
MediaTooth	list of keywords	Supported “roughness” of the media *MediaToothWKV*, MediaNsExtensionPattern	[PWG5100.11] para 11.5.1
MediaType	list of keywords	Supported types of media *MediaTypeWKV*, MediaTypeExtensionPattern	para 3.13.2
MediaWeightMetric	range of int	Supported weight of the media in grams per square meter	para 3.13.9
Any	various		

4.3.2 Service Document Ticket Document Description Capabilities

Document Description Capabilities group Elements provide descriptive information about the Document that can be used in Document Tickets.

Document Description Capabilities may consist of two sequences:

- Elements inherited from the Imaging Document Description Capabilities (Figure 69) These are listed in the table in Table 46
- Service specific Elements (but these are unique to the Print Service and are not covered here.

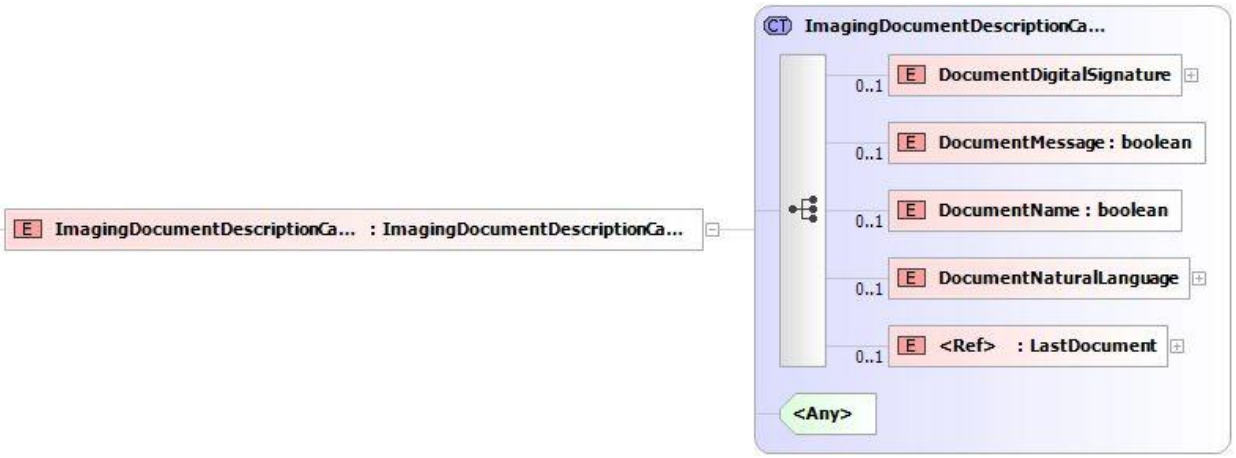

Figure 69 Imaging Document Description Capabilities

Table 46 Imaging Document Description Capabilities Elements

Element	DataType	Description
DocumentDigitalSignature	list of keywords	Allowed values of digital signature
DocumentMessage	boolean	if true, Service supports this feature
DocumentNameDocumentName	boolean	if true, Service supports this feature
DocumentNaturalLanguage	list of keywords	Allowed values of natural language
LastDocument	boolean	if true, Service supports this feature

4.3.3 Service Job Ticket Job Description Capabilities

Job Description Capabilities group Elements provide descriptive information about the Job that can be used in Job Tickets. Job Description Capabilities consists of two sequences:

- Elements inherited from the Imaging Job Description Capabilities super class including Elements such as Job Name and Job Originating User Name (Figure 70.) These are listed in Table 47.
- Service specific Elements, such as the Print Service specific elements shown in Figure 71 and listed in Table 48. The elements listed appear under several Services.

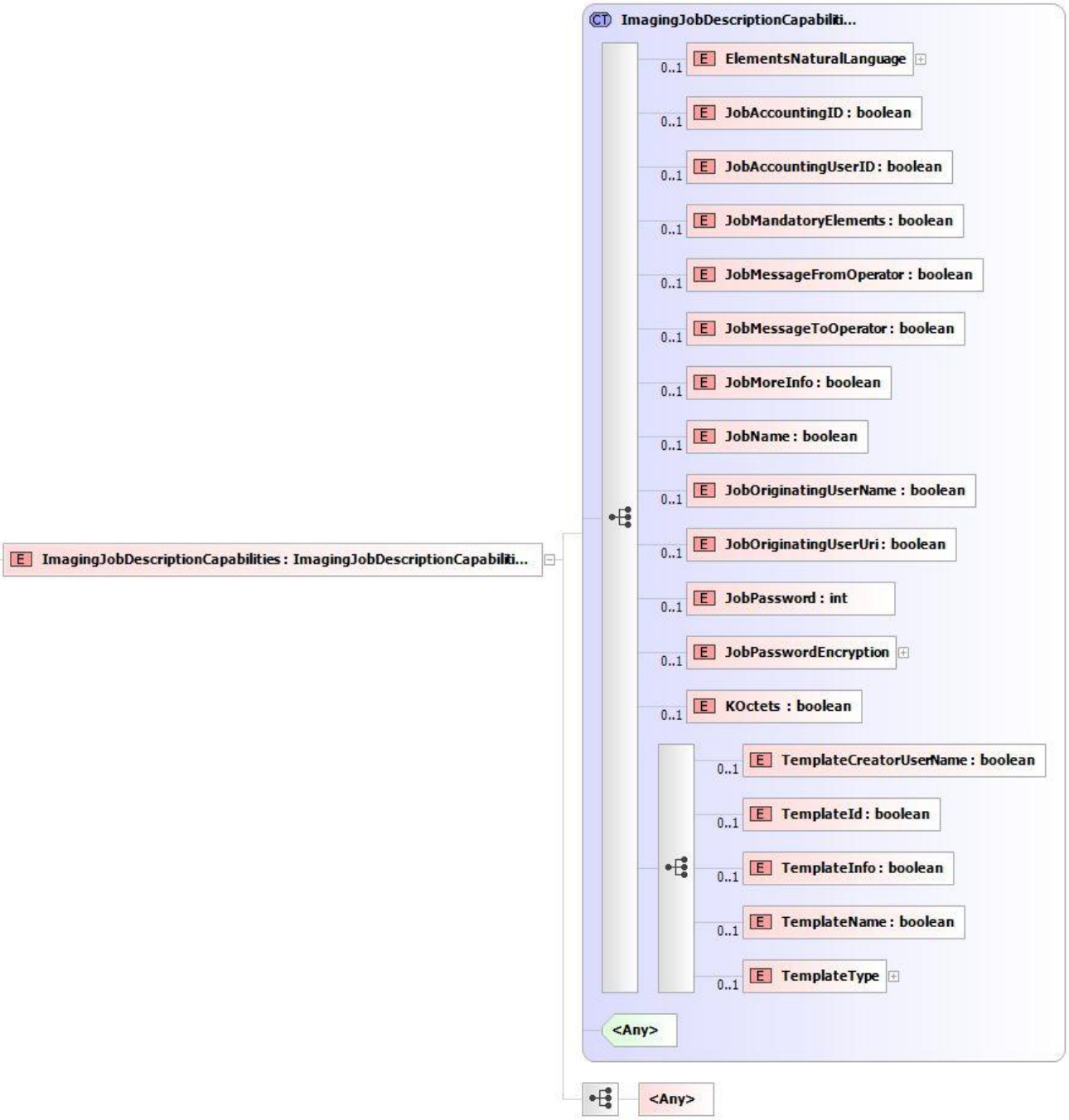

Figure 70 Imaging Service Job Description Capabilities

Table 47 Imaging Service Job Description Capabilities

Element	DataType	Description
ElementsNaturalLanguage	list of keywords	supported natural languages for the Elements with a string syntax (See [RFC3066] *NaturalLanguageWKV*
JobAccountingID	boolean	Service's support for Job Accounts.
JobAccountingUserID	boolean	Service's support for a User supplied Accounting ID associated with Job.
JobMandatoryElements	boolean	Service's support JobMandatoryElements which is an alternative method of encoding MustHonor flagged Elements. This Element is not valid in an XML encoding.
JobMessageFromOperator	boolean	Service's support of messages from the operator for Jobs.
JobMessageToOperator	boolean	Service's support of messages to the operator for Jobs.
JobMoreInfo	boolean	Service's support for a User supplied URI referencing some resource with more information about the Job . [RFC2911] pargraph 4.3.4
JobName	boolean	Service's support of a name for the Job.
JobOriginatingUserName	boolean	Service's support of the name of the user submitting the Job
JobOriginatingUserUri	boolean	Service's support of the URI for the user submitting the Job
JobPassword	int	Indicates service's support for holding a Job until the correct password is entered. Value is the maximum password length accepted.
JobPasswordEncryption	list of keywords	list of password encryption and hash methods supported by the Service. See [DES], [ECC], [AES], [MD2], [MD4], [MD5], [SHA] *JobPasswordEncryptionWKV, KeywordNsExtensionPattern*
KOctets	boolean	Service's support for storing the size of the Job in integral units of 1024 octets.
TemplateCreatorUserName	boolean	Service's support for storing the user who created the template on which the Job Ticket was based.
TemplateId	boolean	Service's support for storing the Id of the template on which the ScanJob Ticket was based.
TemplateInfo	boolean	Service's support for storing information about the template on which Job Ticket was based.
TemplateName	boolean	Service's support for storing the user friendly name of the template on which the Job Ticket was based.
TemplateType	list of keywords	Service's support for storing the type of the template on which the Job Ticket was based. The Service should only support Job and Document Templates. *TemplateTypeWKV*

Figure 71 Print Service Specific Job Description Capabilities

Table 48 Service Specific Job Description Capabilities

Element	DataType	Description
CompressiuonSupplied	list of keywords	*CompressionWKV*, KeywordNsExtensionPattern
DocumentCharsetSuppied	list of keywords	*CharsetWKV*, KeywordNsExtensionPattern
DocumentDigitalSignatureSupplied	boolean	True if supported
DocumentFormatDetailsSupplied	complex	Service's support of Document Format Details (See Figure 72 and Table 49)
DocumentFormatSupplied	list of keywords	*DocumentFormatWKV*, KeywordNsExtensionPattern
DocumentFormatVersionSupplied	boolean	Service's support for Document Format Version
DocumentMessageSupplied	boolean	Service's support for Document Message
DocumentNameSupplied	boolean	Service's support for Document Name
Any (service specific)	various	an extension point for <service>JobDescriptionCapabilities

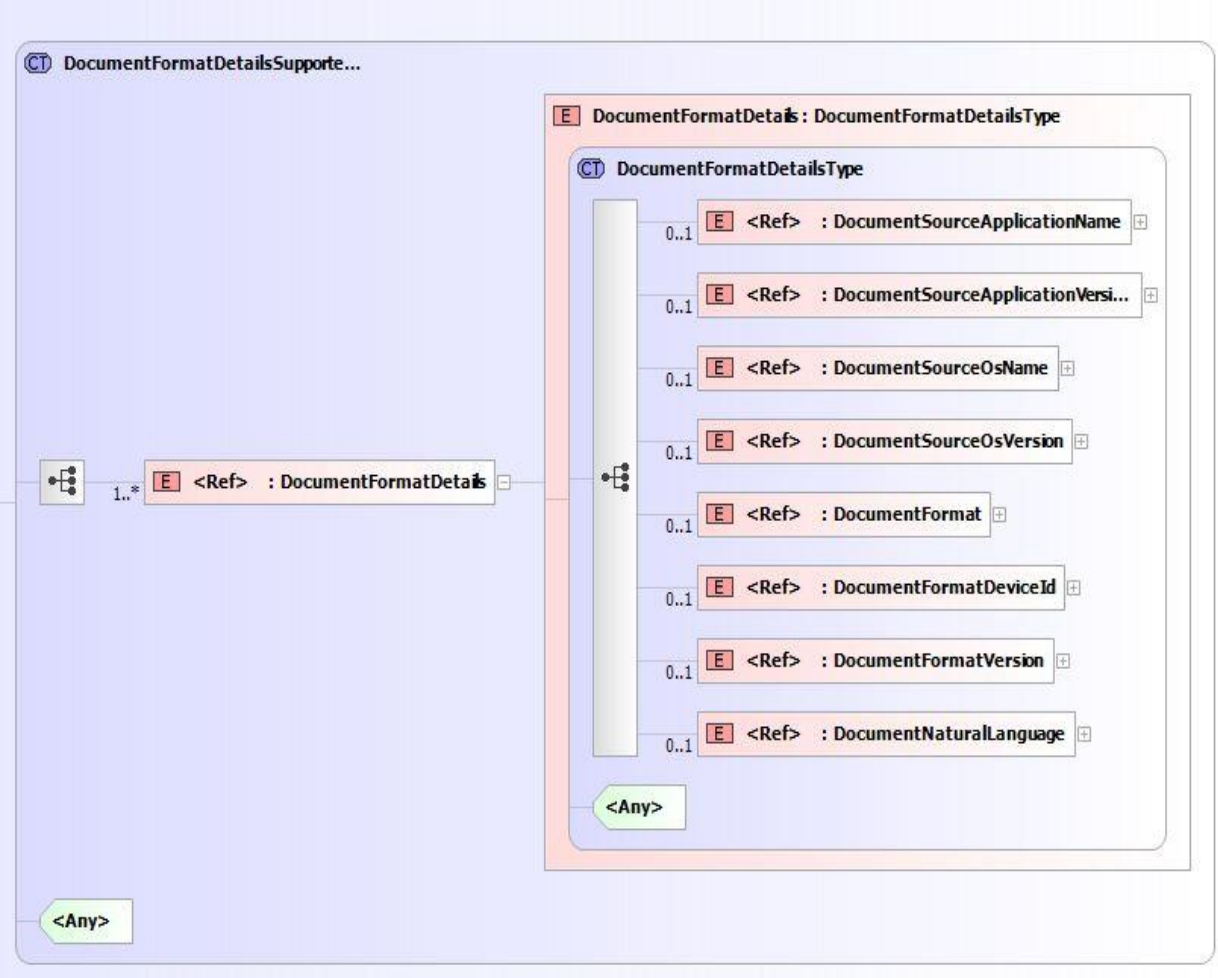

Figure 72 Document Format Details Structure

Table 49 -Document Format Detail Elements

Element	DataType	Description	Reference
DocumentFormatDetailsSupplied	complex	Service's support for Document Format Details (See Figure 72) Following Elements may be supplied for each supported DocumentFormat	
DocumentSourceApplicationName	string	Name of the application that created the document	[PWG5100.5] para 9.1.13
DocumentSourceApplicationVersion	string	Version of the application that created the document	[PWG5100.5] para 9.1.13
DocumentSourceOsName	keyword	Name of the operating system on which the document was generated para	[PWG5100.5] 9.1.13
DocumentSourceOsVersion	string	Version of the operating system on which the document was generated para	[PWG5100.5] 9.1.13
DocumentFormat	keyword	Document format (i.e., PDL) for this Document	[RFC2911] para 3.2.1.1
DocumentFormatDeviceId	string	Type of device for which the document was formatted	[PWG5100.5] para 9.1.13
DocumentFormatVersion	string	Level or version of the DocumentFormat	[PWG5100.5] para 9.1.16
DocumentNaturalLanguage	keyword	Primary Natural Language of this Document	[RFC2911] para 3.2.1.1

4.3.4 Service Job Ticket Job Processing Capabilities

The Job Processing Capabilities group Element provides information about the Job processing Elements that can be used in Job Tickets. “Capabilities” values are often just a Boolean indicating whether or not the identified Job Processing Element is supported. In other instances, it may be a list of supported values.For descriptions of the Job Processing Elements themselves, see paragraph 5.2.3.

Job Processing Capabilities consists of two sequences:

- Elements inherited from the Imaging Service super class Job Processing Capabilities Type, (Figure 73) including Elements such as Job Hold Until. These are listed in Table 50.
- Service specific Elements. The FaxOut Service’s Service Job Processing Capabilities shown in Figure 74, and the Copy Service Job Processing Capabilities shown in Figure 75 and Figure 76 include Service-specific capabilities applicable to several Services. Table 51 lists these Job Processing Capabilities Elements as well as some others that are not included in the example Service Schema illustrations.

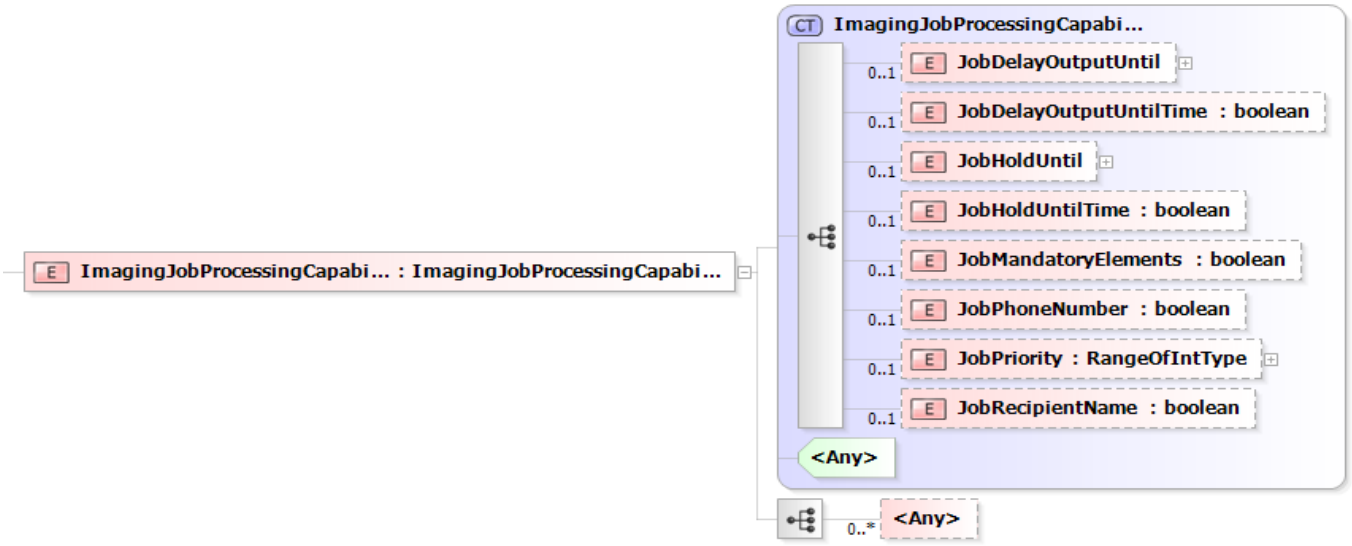

Figure 73 Imaging Service Job Processing Capabilities

Table 50 Imaging Service Job Processing Capabilities

Element	DataType	Description or *Keyword*	Reference
JobDelayOutputUntil	list of keywords	the named events for releasing the delayed output	[PWG5100.11] para 7.4
JobDelayOutputUntilTime	boolean	Service allows a Job Ticket to specify a Date and Time when the output delay is to expire.	[PWG5100.11] para 7.5
JobHoldUntil	list of keywords	the named events for releasing the held Job. *JobHoldUntilWKV*	[RFC2911] para 4.2.2
JobHoldUntilTime	boolean	Service allows a Job Ticket to specify a Date / Time when a Job that is on hold will be released.	[PWG5100.11] para 7.6
JobMandatoryElements	boolean	The Service support sJobMandatoryElements.	[PWG5100.5] para 8.1
JobPhoneNumber	boolean	The Service supports storing the contact phone number for the Job	[PWG5100.11] para 7.7
JobPriority	range of int		[RFC2911] para 4.2.1
JobRecipientName	boolean	The Service supports storing the name of the recipient of a Job	[PWG5100.11] para 7.8
Any	various	Extension point for JobProcessingCapabilities	

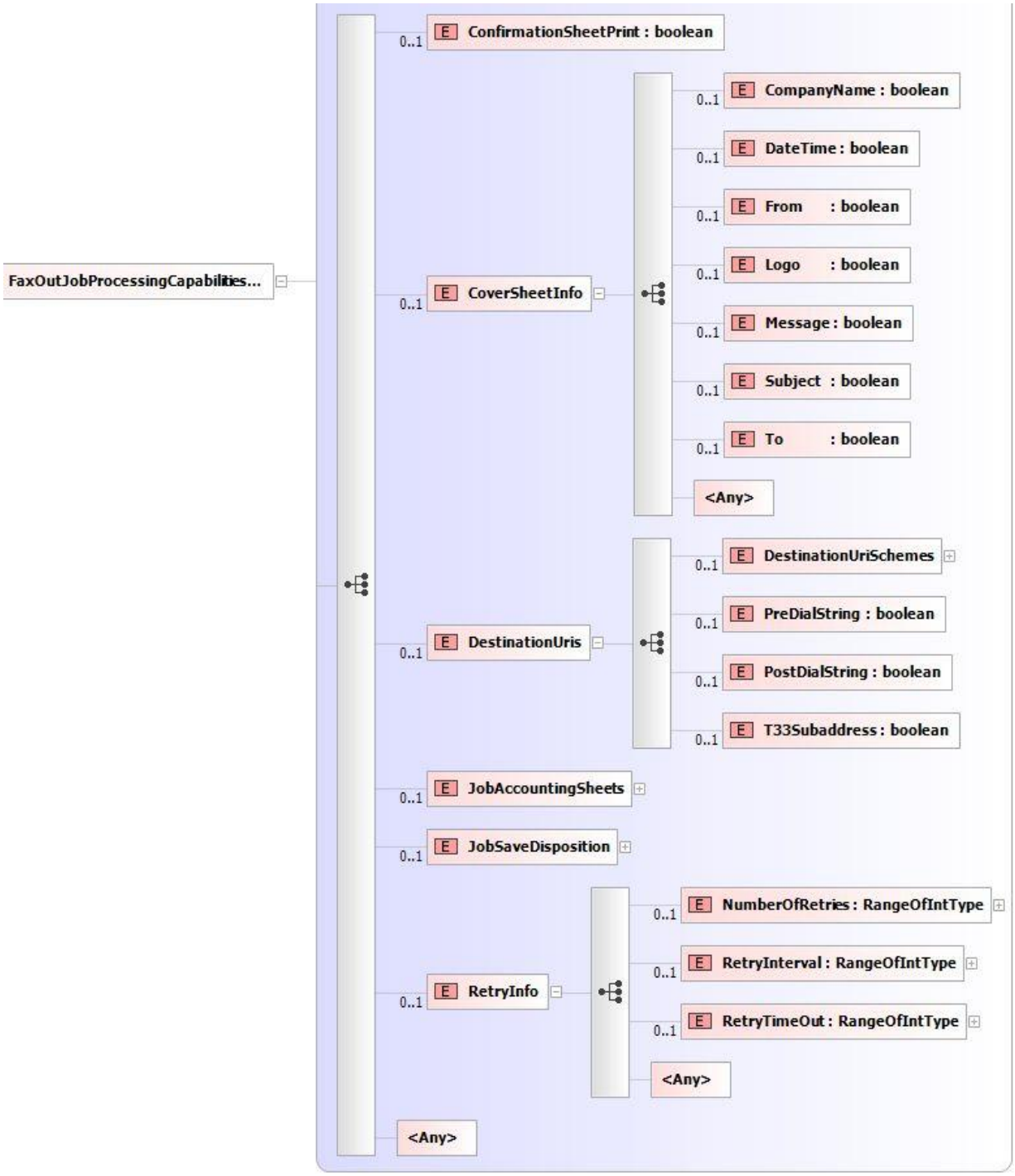

Figure 74 Representative Service-Specific JobProcessingCapabilities Elements (FaxOut Service Examples)

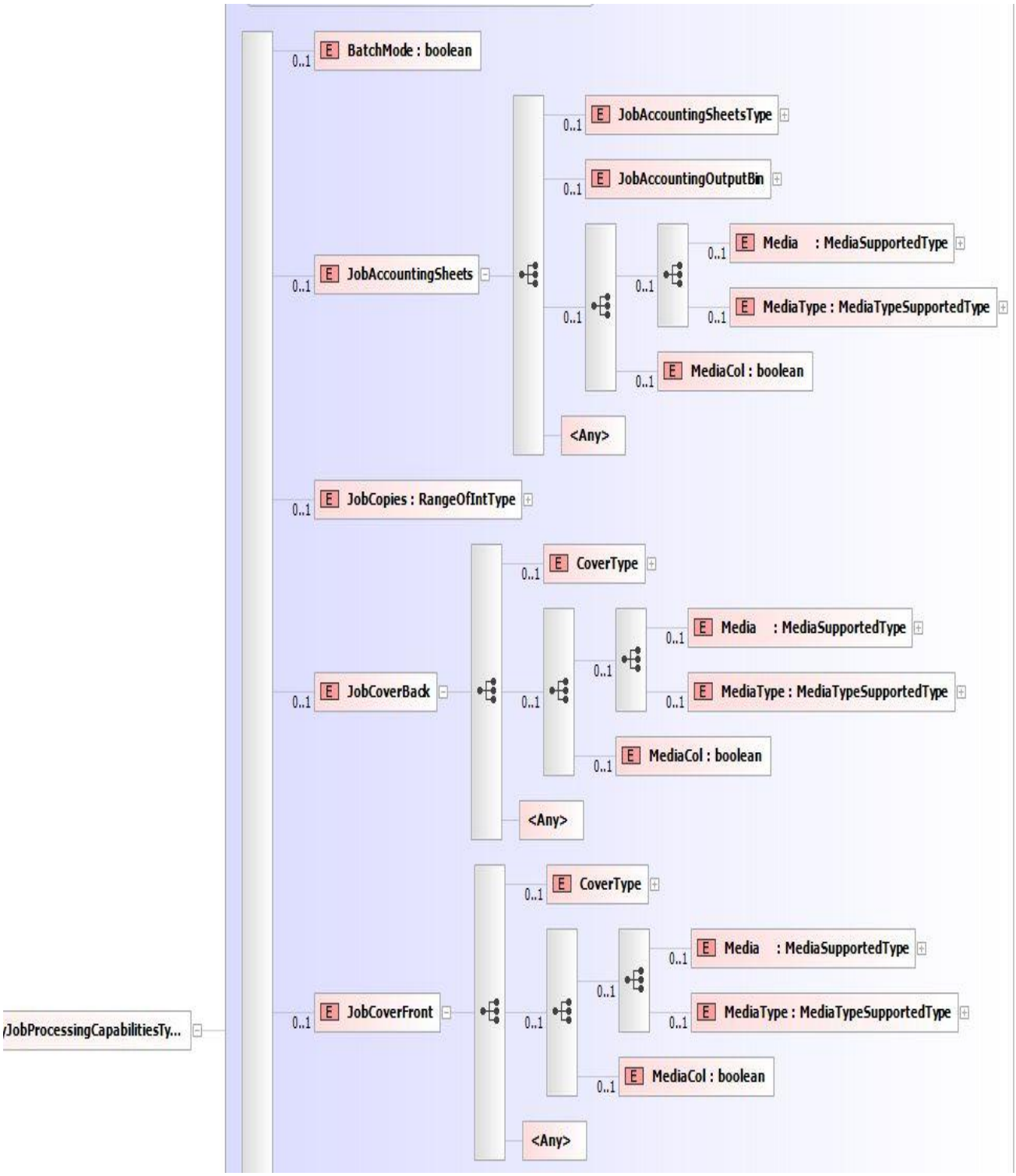

Figure 75 Representative Service-Specific JobProcessingCapabilities Elements (Copy Service Examples) Sheet 1

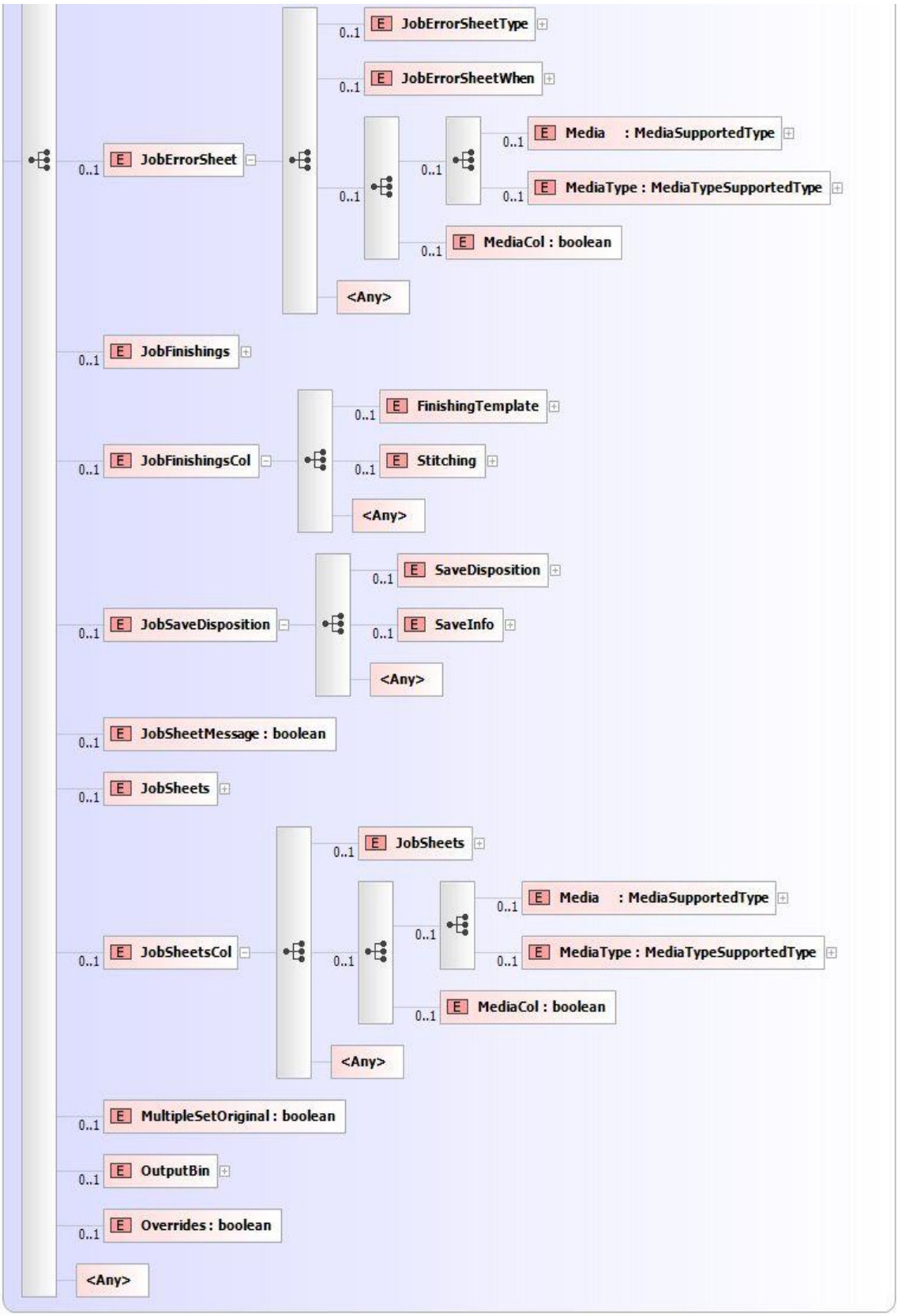

Figure 76 Representative Service-Specific JobProcessingCapabilities Elements (Copy Service Examples) Sheet 2

Table 51 Service-Specific JobProcessingCapabilities Elements

Element	DataType	Description or *Keyword*	Reference
BatchMode	boolean	True = Service supports the BatchMode element (Figure 75)	[PWG5108..2] para 7.1.2.3.5
ComfirmationSheetPrint	boolean	True = Service supports the ComfirmationSheetPrint element. The format of the confirmation sheet is implementation specific. (Figure 74)	
CoverSheetInfo	complex	(Figure 74)	
CompanyName	boolean	True = Service supports the Element conaining the user-supplied name of the company for the cover sheet.	
DateTime	boolean	True = Service supports the Element containing the user-supplied date and time for the cover sheet.	
From	boolean	True = Service supports the Element containing user-supplied name of the person group or entity for the cover sheet	
Logo	boolean	True = Service supports the Element containing user-supplied URL pointing to image used as the logo on the cover sheet.	
Message	boolean	True = Service supports the Element containing user-supplied message on the cover sheet.	
Subject	boolean	True = Service supports the Element containing user-supplied subject line on the cover sheet.	
To	boolean	True = Service supports the Element containing user-supplied addressee line on the cover sheet.	
Any	various	Extension point for CoverSheetInfo	
DestinationUris	Complex	feature is supported (Figfure 74)	
DestinationUriSchemes	list of strings	URI Schemes supported that can be used to transfer a Digital Document to its Destination. See	DestinationUriSchemes [RFC4395]
PreDialString	boolean	True = Service supports the Element defining a Dial string to be entered before the DestinationUri is applied. See 5.2.3.1.	
PostDialString	boolean	True = Service supports the Element defining a Dial string to be entered after the DestinationUri is applied. See 5.2.3.1.3.	
T33Subaddress	boolean	True = Service supports the Element allowing T33 subaddressing. See 5.2.3.1.4.	[RFC3192]
DocumentOutputMode	list of keywords	(Scan, etc) *DocumentOutputModeWKV*, KeywordNsExtensionPattern	[PWG5108.2] para 8.1.3.3.6
JobAccountingSheets	complex	(Figure 75)	[PWG5100.3] para 3.8
JobAccountingSheetsType	list of keywords	*JobAccountingSheetsTypWKV*	[PWG5100.3] para 3.8.1
JobAccountingOutputBin	list of keywords	JobAccountingOutputBin	[PWG5100.3] para 3.8.3
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	list of keywords	*MediaTypeWKV* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	boolean	Indicates if MediaCol is supported for JobAccountingSheets	[PWG5100.3] para 3.13
Any	various	Extension point for JobAccountingSheets	
JobCopies	range of int	Numbers of copies supported (Figure 74)	[PWG5100.7] para 4.1.1
JobCoverBack	complex	(Figure 75)	[PWG5100.7] para 4.1.2
CoverType	list of keywords		[PWG5100.3] para 3.1.2

Element	DataType	Description or *Keyword*	Reference
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	list of keywords	*MediaTypeWKV* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	boolean	Indicates if MediaCol is supported for JobCoverBack	[PWG5100.3] para 3.13
Any	various	Extension point for JobCoverBack	
JobCoverFront	complex	(Figure 75)	[PWG5100.7]para4.1.3
CoverType	list of keywords		[PWG5100.3]para3.1.2
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	list of keywords	*MediaTypeWKV* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	boolean	*Indicates if MediaCol is supported for JobCoverFront*	[PWG5100.3] para 3.13
Any	various	Extension point for JobCoverFront	
JobErrorSheet	complex	(Figure 76)	
JobErrorSheetType	list of keywords	*JobErrorSheetTypeWKV*	
JobErrorSheetWhen	list of keywords	*JobErrorSheetWhenWKV*	
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	list of keywords	*MediaTypeWKV* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	Boolean	*Indicates if MediaCol is supported for JobErrorSheet*	[PWG5100.3] para 3.13
Any	various	Extension point for JobErrorSheet	
JobFinishings	list of keywords	*JobFinishingsWKV* (Figure 76)	[RFC2911] para 4.2.6 [PWG5100.1] para 2
JobFinishingsCol	complex	(Figure 76)	[PWG5100.3] para 3.2
FinishingTemplate	list of string	Allowed values	[PWG5100.3] para 3.2.1
Stitching	complex		[PWG5100.3] para 3.2.2
StitchingReferenceEdge	list of keywords	*StitchingReferenceEdgeWKV*	[PWG5100.3] para 3.2.2.1
StitchingOffset	range of int		[PWG5100.3] para 3.2.2.2
StitchingLocations	range of int		[PWG5100.3] para 3.2.2.3
Any	various	Extension point for Stitching	
Any	various	Extension point for JobFinishingsCol	
JobSaveDisposition	complex	(Figure 76)	[PWG5100.11] para 7.9
SaveDisposition	list of keywords	*SaveDispositionWKV*	[PWG5100.11] para 7.9.1.1
SaveInfor	complex		[PWG5100.11] para 7.9.1.2
SaveDocumentFormat	list of keywords	SaveDocumentFormatType	[PWG5100.11] para 7.9.1.2.3.3
SaveLocation	list of strings	URI Schemes supported that can be used to transfer a Digital Document to its Destination. See [RFC4395]	[PWG5100.11] para 7.9.1.2.3.1
SaveName	boolean		[PWG5100.11] para 7.9.1.2.3.2
Any	various	Extension point for SaveInformation	
Any	various	Extension point for JobSaveDisposition	
JobSheetMessage	boolean	feature is supported (Figure 76)	[PWG5100.3] para 3.12
JobSheets	list of keywords	(Figure 76) *JobSheetsWKV*, KeywordNsExtensionPattern	[RFC2911] para 4.2.3 [PWG5100.3] para 6.2
JobSheetsCol	complex	Figure 76	[PWG5100.3] para 3.11

Element	DataType	Description or *Keyword*	Reference
JobSheets	list of keywords	*JobSheetsWKV*, KeywordNsExtensionPattern	[RFC2911] para 4.2.3 [PWG5100.3] para 6.2
Media	list of keywords	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	list of keywords	*MediaTypeWKV* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	boolean	Indicates if MediaCol is supported for JobAccountingSheets	[PWG5100.3] para 3.13
Any	various	Extension point for JobSheetsCol	
MultipleDocumentsHandling	list of keywords	*MultipleDocumentsHandlingWKV,* KeywordNsExtensionPattern	[RFC2911] para 4.2.4
MultipleSetOriginal	boolean	feature is supported (Figure 74)	[PWG5108.2] para 7.1.2.3.7
OutputBin	list of keywords	(Figure 74) *OutputBinWKV*, KeywordNsExtensionPattern	[PWG5100.2] para 2.1 [PWG5100.5] para 8.1
OutputDevice	list of string	Allowed values (Print)	[PWG5100.7] para 4.2.1 [PWG5100.5] para 8.1
Overrides	llst of keywords	List of JobTicket elements that can be used in an override for this service	[PWG5100.6] para 4.1
PagesPerSubset	boolean	feature is supported (Print)	[PWG5100.4] para 5.3
RetryInfo	complex	Supported capabilities of elements used to control document transmission timeouts	
NumberOfRetries	range of int	Allowed range of values for number of attempts to retransmit a document in the event of transmission failure s, before aborting the job	
RetryInterval	range of int	Allowed range of values for time duration in seconds between transmission attempts	
RetryTimeout	range of int	Allowed range of values for time duration in seconds before terminating a transmission attempt when no document data is being sent.	
Any(service specific)	various	an extension point for RetryInfo	
Any (service specific)	various	an extension point for <service> JobProcessingCapabilities	

4.4 Service Configuration

The Service Configuration Element for each service contains the subset of the Subunits identified in the System Configuration (Chapter 3) that may be necessary for that particular service. The Elements comprising each identified Subunit are described in Chapter 3.

The individual Service specific specifications define the subset of System Configuration Subunits applicable to each Service.

4.5 Service Defaults

The Service Defaults (Figure 77) can include a Default Document Ticket (Figure 78) and a Default Job Ticket (Figure 79.) These default tickets provide the values that will be used for Elements values omitted in the supplied Tickets. The processing instructions are not bound to the Job until the Job is actually processed nor are they bound to the Document until the Document is processed. The values from the Default Tickets are not copied to the Job's Job Ticket or Document Ticket(s). If the Job Receipt is supported, the combined Elements from the user supplied Job Ticket and the applied values from the Default Job Ticket are copied to the Job Receipt.

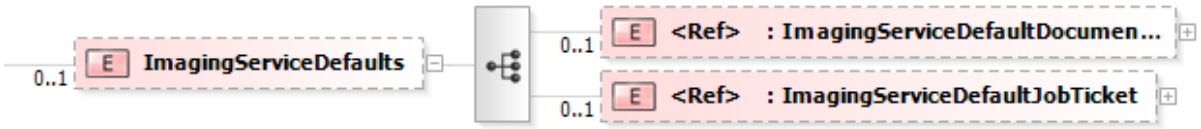

Figure 77 Imaging Service Defaults

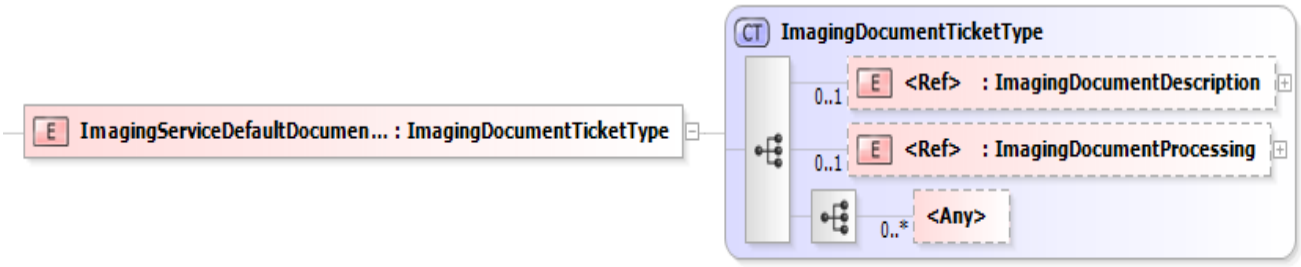

Figure 78 Basic Document Ticket Defaults

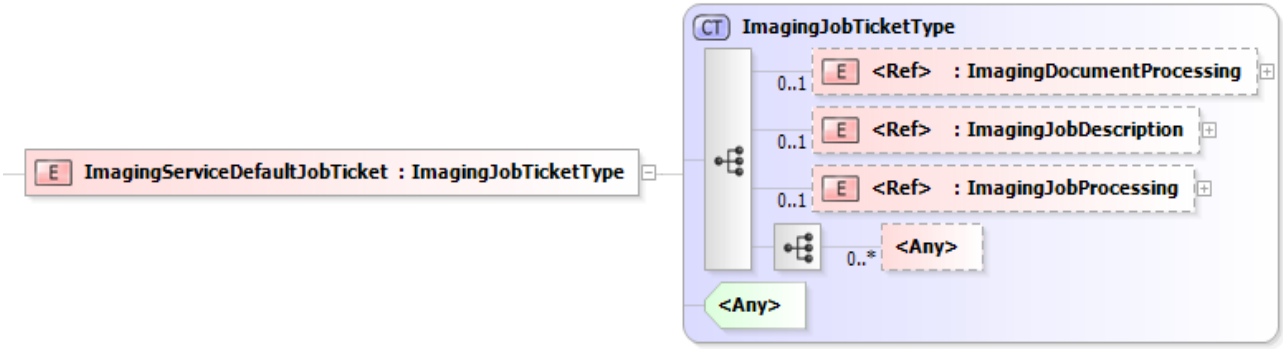

Figure 79 Imaging Service Job Ticket Defaults

The Default Job Ticket may contain all of the Elements in a Job Ticket, as described in section 0. The Default Document Ticket may contain all of the Elements of a supplied Document Ticket, as described in Section 6.2

4.6 Service Description

The Service Description Elements of a Service provide descriptive information; that is, the values of the Elements can be administratively set and/or can be modified directly or indirectly through an operation. Service Description consists of two sequences.

- Elements inherited from the Imaging Service super class. These are represented in Figure 80 with the constituent Elements listed in Table 52.
- Service specific Elements. Service-specific Elements for the FaxOut Service, which are shared with some other Services, are represented in Figure 81. Service-specific Description Elements used by more than one Service are listed in Table 53.

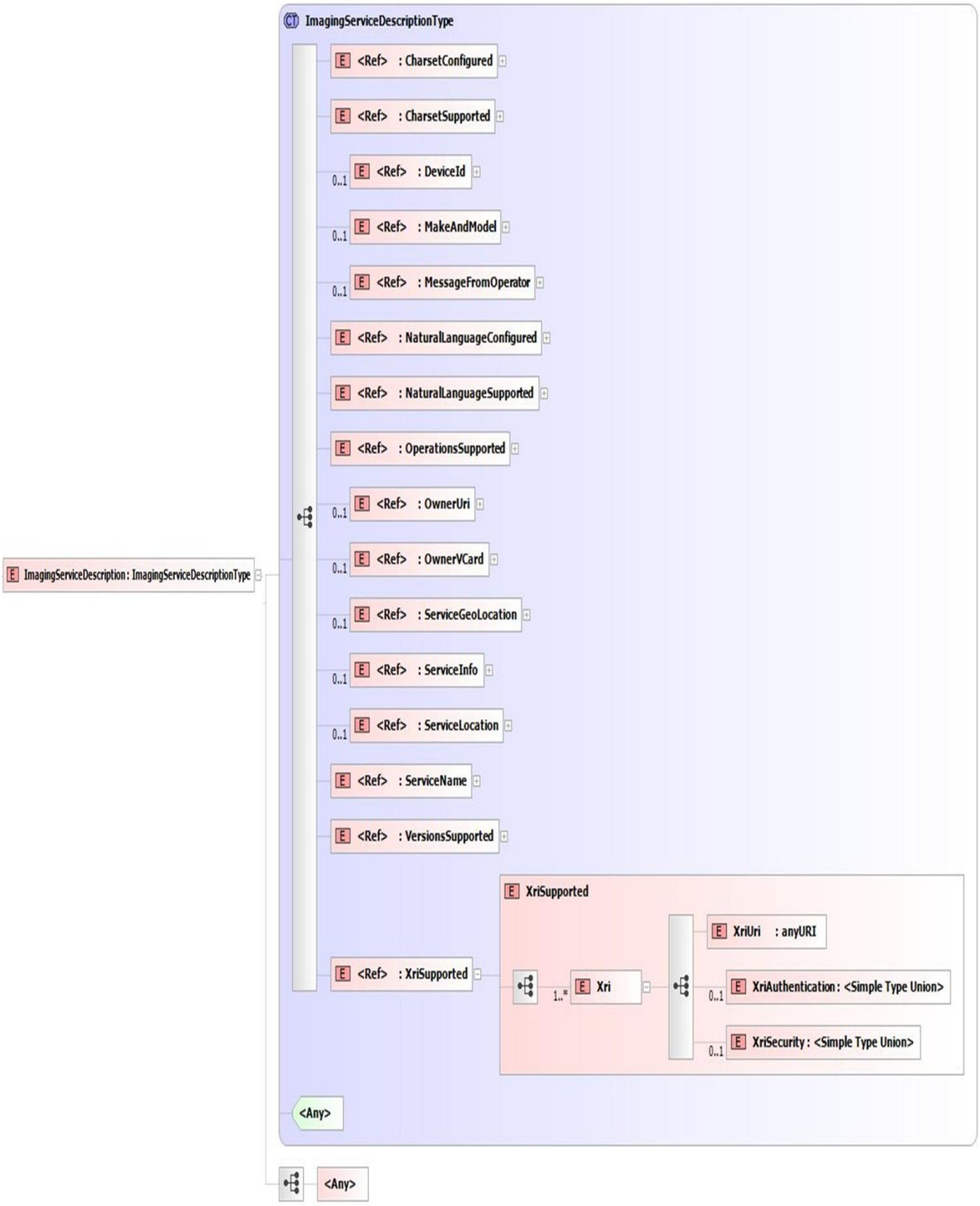

Figure 80 Imaging Service Description

Table 52 Service Description Elements

Element	DataType	Description or *Keyword*	Reference
CharsetConfigured	keyword	*CharsetWKV*	[RFC2911] Para 4.3.19
CharsetSupported	list of keywords	*CharsetWKV*	[RFC2911] Para 4.3.19
DeviceId	string	IEEE 1284 Device ID	[PWG 5105.1] Para 11.1
MakeAndModel	string		[RFC2911] Para 4.4.9
MessageFromOperator	string		[RFC2911] Para 4.4.25
NaturalLanguageConfigured	keyword	*NaturalLanguageWKV*	[RFC2911] Para 4.4.19
NaturalLanguageSupported *	list of keywords	*NaturalLanguageWKV*	[RFC2911] Para 4.4.20
OperationsSupported	list of keywords	*OperationsSupportedWKV*	[RFC2911] Para 4.4.15
OwnerURI	anyUri	URI that is an authoritative identifier (e.g., a 'mailto:' URI) of the authenticated Owner of this Service instance.	[RFC3986]
OwnerVCard	string	A MIME vCard that contains contact information (e.g., email, postal, telephone, etc.) for the authenticated Owner of this Service instance.	[RFC2426]
ServiceGeoLocation	anyUri		[RFC5870]
ServiceInfo	string		[RFC2911] Para 4.4.6
ServiceLocation	string		[RFC2911] Para 4.4.5
ServiceName	string		[RFC2911] Para 4.4.4
VersionsSupported	list of strings		[RFC2911] Para 4.4.14
XriSupported	complex	Service transport ednpoint	
XriUri	anyURI		[RFC2911] Para 4.4.1
XriAuthentication	list of keywords	*UriAuthenticationWKV*	[RFC2911] Para 4.4.2
XriSecurity	list of keywords	*UriSecurityWKV*	[RFC2911] Para 4.4.3
Any	various	Extension point for ServiceDescription	

Note: * NaturalLanguageSupported maps to generated-natural-language-supported in IPP

4.6.1 OwnerURI

This Service Description element is a URI [RFC3986] that is an authoritative identifier (e.g., a 'mailto:' URI) of the authenticated Owner of this Service instance. This element MAY be usable to deliver notifications to the Owner. The abstract syntax for this element is a string.

4.6.2 OwnerVCard

This Service Description element is a MIME vCard [RFC2426] that contains contact information (e.g., email, postal, telephone, etc.) for the authenticated Owner of this Service instance. This element SHOULD be usable to deliver notifications (e.g., security alerts) to the service owner. The abstract syntax for this element is a string.

4.6.3 MultipleOperationTimeoutAction

This Service Description element specifies what action the service should take when a job times out. The data type for this element is keyword. This element can be extended by adding new keywords. The defined keywords are:

- 'Abort' – the job is closed and aborted,
- 'Hold' – The job is closed and put on indefinite hold,
- 'Process' – the job is closed and queued for processing.

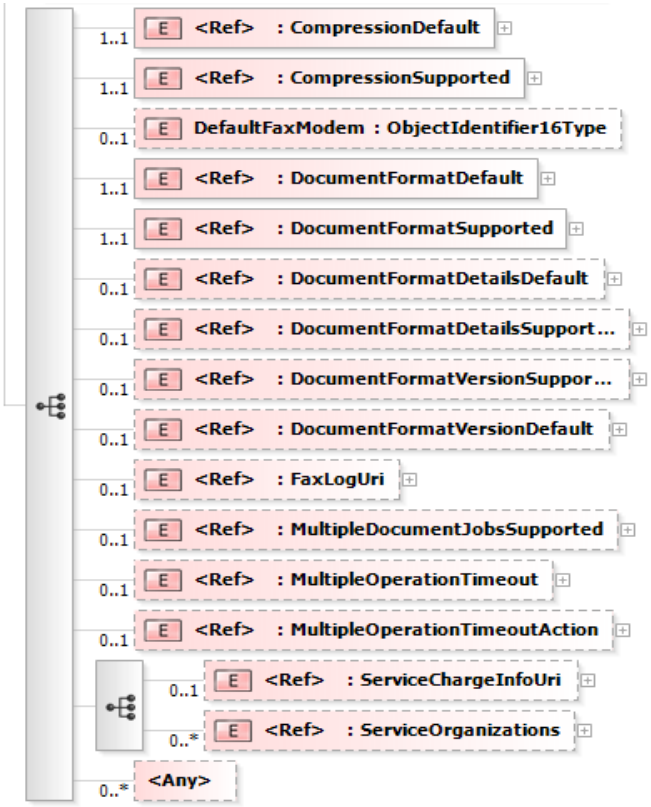

Figure 81 Representative Service-Specific Service Description (FaxOut)

Table 53 Service-Specific Service Description Elements

Element	DataType	Description	Reference
CompressionDefault	keyword	Default Compression algorithm assumed to be used on the Document Data	[RFC2911] Para 4.4.32
CompressionSupported	list of keywords	Identifies the set of supported compression algorithms for Document content	[RFC2911] Para 4.4.32
DocumentFormatDefault	keyword	Default document format (i.e., PDL) assumed for a Document	[RFC2911] Para 3.2.1.1 [PWG5100.5] Para 9.1.12
DocumentFormatSupported	list of keywords	Identifies the set of supported document format s	[RFC2911] Para 3.2.1.1 [PWG5100.5] Para 9.1.12
DocumentFormatDetailsDefault	complex	the member attributes of Doc umentFormatDetails that the Service supports See Figure 82 and Table 54	[PWG5100.7] Para 7.6 –
DocumentFormatDetailsSupported	list of keyword	Default Compression algorithm assumed to be used on the Document Data	[RFC2911] Para 4.4.32
DocumentFormatVersionSupported	list of strings	levels or versions of the DocumentFormats that the Service will accept	[PWG5100.7] Para 7.8
DocumentFormatVersionDefault	string	level or version of the DocumentFormat that the Service will assume when none is provided by the client	[PWG5100.7] Para 7.7
MultipleDocumentJobsSupported	boolean		

Element	DataType	Description	Reference
MultipleOperationTimeout	int	Identifies the minimum time (in seconds) that the service will wait between actions on an open job before timing out	[RFC2911] Para 4.4.31
MultipleOperationTimeoutAction	keyword	value indicates what action the service should take when a job times out. This element can be extended by adding new keywords.Defined keywords are: 'Abort' – the job is closed and aborted, 'Hold' – The job is closed and put on indefinite hold, 'Process' – the job is closed and queued for processing.	
ServiceChargeInfoUri	anyURI		
ServiceOrganizations	complex		
ServiceOrganization	string		
ServiceOrganizationUnit	string		
Any	various	Extension point for ServiceOrganizations	
Any	various	Extension point for <service>ServiceDescription	

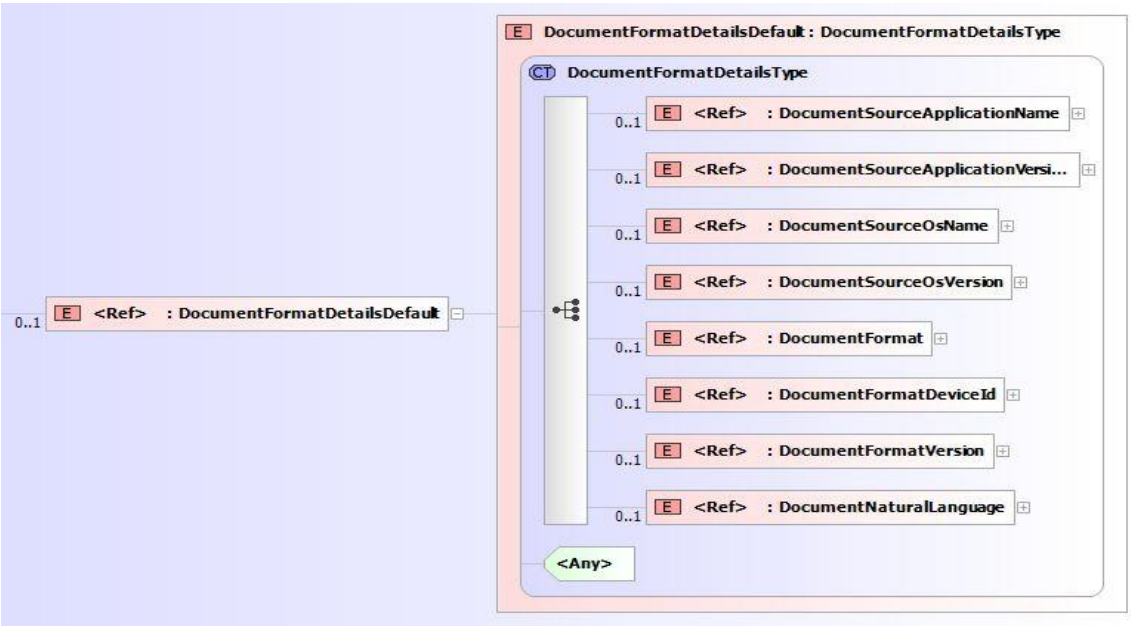

Figure 82 Detail of DocumentFormatDetailsDefault

Table 54 Constituent Elements of DocumentFormatDetailsDefault

Element	DataType	Reference
DocumentFormatDetailsDefault	complex	[PWG5100.7] section 7.6
DocumentSourceApplicationName	string	
DocumentSourceApplicationVersion	string	
DocumentSourceOsName	keyword	
DocumentSourceOsVersion	string	
DocumentFormat	keyword	
DocumentFormatDeviceId	string	
DocumentFormatVersion	string	
DocumentNaturalLanguage	keyword	

4.7 Service Status

The Status Elements provide state information for the Service. The Elements are maintained by automata and cannot be directly set. The Element values may be modified indirectly through an operation. For example, a <*service*> Pause operation may result in the change of the State and State Reasons Elements.

Service Status Elements consists of two sequences.

- Elements inherited in the Service Status type from the Imaging Service super class including Elements such as Id and State. This includes work counters specific to output and input of the service; e.g.,Images vs Impressions; Jobs vs Resources. These are listed in Table 55 and illustrated in Figure 83.
- Service specific Elements. The common Service-specific Elements for each image processing Service, are the Service-specific counters, as identified in Table 56.

The Service-specific elements are differentiated by each Service having a complex <service>Counters Element which includes the counters applicable to that Service. Counters are discussed separately in Chapter 8, Counters & Timers.

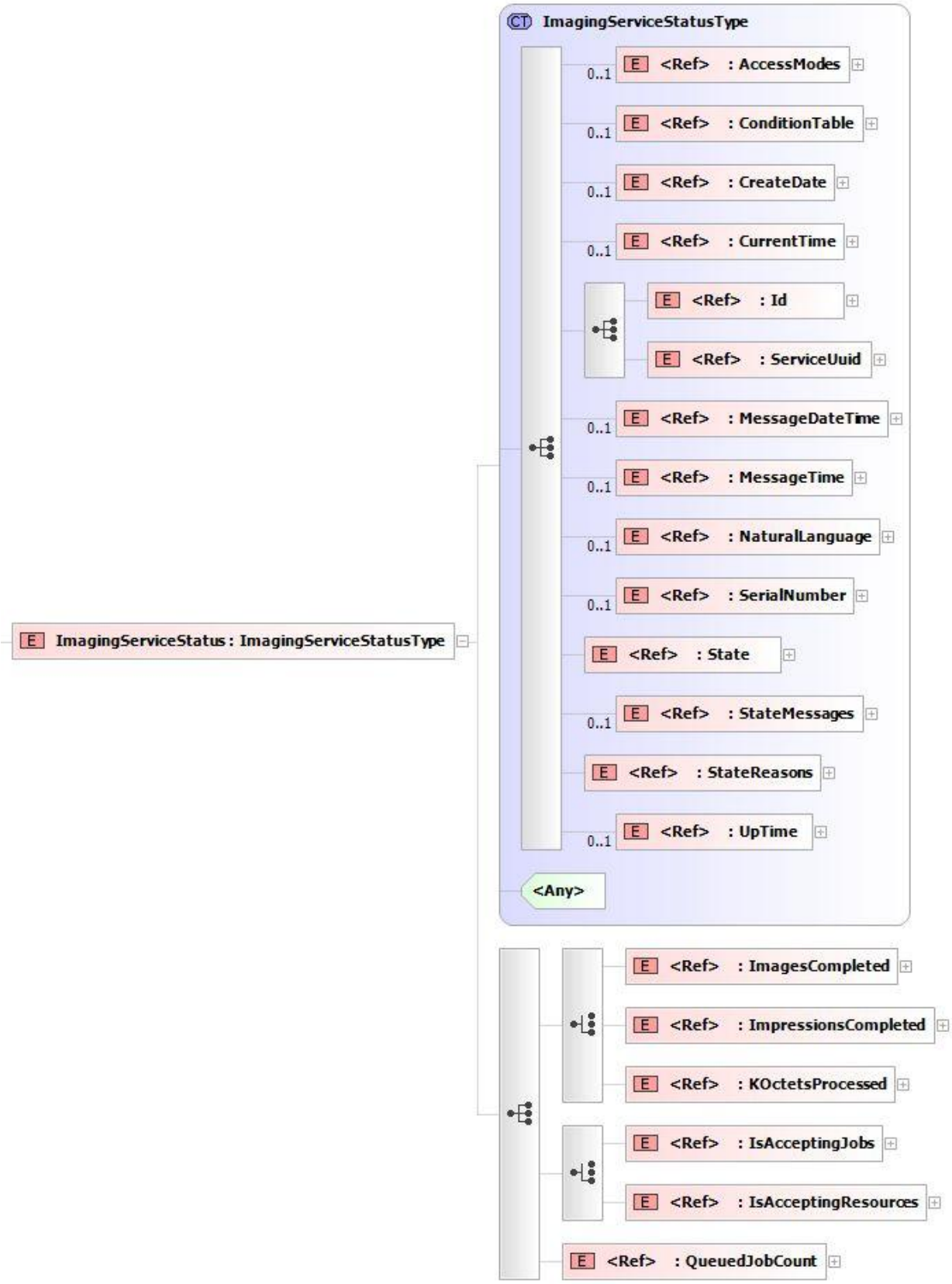

Figure 83 Imaging Service Status

Table 55 Service Status Elements

Element	DataType	Description
AccessModes	list of keywords	basic access control policy for authenticated users. This Element corresponds to the access mode property of a POSIX file and specifies the basic access control policy for the Service object, as set by the Owner. The AccessMode Element takes precedence over any externally specified access policy. ObjectAccessModeWKV
ConditionTable	complex	additional information on the current and past state of Services and Subunits. See 4.7.1
CreateDate	dateTime	date and time that the Service was created
CurrentTime	dateTime	current date and time according the Service's internal clock
ID	int	a32 bit Object Identifier for the Scan Service instance. It is unique withinthe containing System.
ServiceUuid	anyUri	An urn::uuid unique URI value [RFC4122]
MessageDateTime	dateTime	
MessageTime	int	
NaturalLanguage	string	local language currently used by the Service. This is the language used unless the operation specifies a RequestedNaturalLanguage(example values: en-us, fr, de, ja) See [RFC3066].
SerialNumber	string	the serial number of the device hosting the service.
State	keyword	current state of scanning service. The state is a unification of the service states from IPP and the Host Resource MIB. See [RFC2911] and [RFC2790]. See paragraph7.2.1 (*ObjectStateWKV*)
StateMessages	list of string	information about the Service State and StateReasons in human readable text. If the Service supports this Element, it must be able to generate the messages in any of the natural languages supported by the Service.
StateReasons	list of keywords	additional detail about the service's state. The keywords are extensible. The standard keyword values are defined in paragraph 4.4.12 of [RFC2911] and paragraph 4.4.3.1 of [WS-Scan]. (*StateReasonsWKVs*)
UpTime	int	Time that the Service has been in the Operational UP state, in seconds.
ImagesCompleted	int	Lifetime count count of what the Service has completed. See Counter Spec [PWG5106.1 Images or impressions are in units of 1; KOctets are in units of 1024 octets.
ImpressionsCompleted	int	
KOctetsProcessed	int	
IsAcceptingJobs	boolean	If True, Service is currently able to accept CreateJob operation. Method of configuring the value for this Element is implementation-specific, e.g., local console, web page.
IsAcceptingResources	boolean	an extension point for vendor differentiation and implementation specific extensions while maintaining interoperability.
QueuedJobCount	int	Number of Jobs this Service has currently created but not yet completed

Table 56 Service-Specific Service Status Elements

Element	DataType	Description
ServiceCounters	complex	the counters for the amount of work performed by the Service, timers covering utilization and monitoring information covering errors, warnings, traffic, Job counts and configuration changes. See Chapter 8 and the PWG Counter Spec. [PWG5106.1]

4.7.1 Condition Table

The Condition Table Elements provide additional current and past state information for the Service and its associated Subunits. Note that the System's Condition Table provides a global view of all conditions. The entries are maintained by automata and cannot be modified by users or administrators. This Element's information maps to the Printer MIBs Alert Table [RFC3805]. One improvement is that a history may be kept for when Alerts (i.e., Conditions) are corrected.

Note that Condition Elements consists of two sequences.

- A mandatory list of active conditions
- An optional Condition History.

Note that the only difference between the data types for Active Condition and Condition History entries is that Condition Histories contain an Element containing the time the condition was cleared. Condition Table Elements are shown in Figure 84 and are described in Table 57.

Figure 84 Condition Table

Table 57 Condition Table Elements

Element	DataType	Description
Component	complex	
Group	keyword	The type of Subunit/service for this condition
Name	string	Instance identifier for the Subunit/service for this condition
ConditionId	int	Condition instance identifier
Name	keyword	The Condition Name: *StateReasonsWKV*, KeywordNsExtensionPattern
Severity	keyword	Severity level of the condition: Informational, Warning, Critical
Time	dateTime	Date and time the condition occurred
ClearTime	dateTime	Date and time the condition was removed

4.7.2 Relationship among Service State, Service State Reasons, Condition Table and Service State Message Elements

There are a number of related elements in the Service Status group that describe the state of a Service, with the primary indicator being the Service State element. This element has a well-defined set of values that cannot be extended. The transitions between the values of Service State are also well defined. (See section 7.1 and 7.2.1). This permits interoperable interpretation of the state of a service by independently developed automata.

However, although the Service State element allows for a high level representation of the state of a Service, it does not provide enough detail to understand the reason for the Service being in a particular state or what work (if any) is being performed. Therefore, Service State is augmented by the Service State Reasons element. Service State Reasons has a well-defined base set of values that can be extended. Extensions are permissible by the PWG, by a vendor, or even an instance of a Service. This permits interoperable interpretation of the state reason for a service by independently developed automata with the caveat that vendor or site specific values will be treated as 'Other' values.

The Service State Reasons provides additional detail for the current state a Service. For example a Service might be in a ‘Stopped’ state as indicated by the value of Service State, but Service State Reasons might provide additional detail for the Service’s state. Service State Reasons could contain a value such as ‘Media Empty’. Note that both Service State and Service State Reasons are needed to interpret the affect of a Service State Reason on the state of a Service. For example. a “Media Empty” Service State Reason element value can indicate either a warning condition for a Service or the cause for the service to move from a ‘Processing’ state to a ‘Stopped’ state. (i.e., the Service will only move to a Stopped state if the Service requires the media previously available in that tray to continue processing the current job.)

The information inherent in the Service State Reason value may be very limited; for example ‘Media Empty’ does not identity what tray is empty. Additional information may be available in the Condition Table element of Service Status (4.7.1). The Condition Table element contains two tables that represent the current conditions and the previous conditions for the Device hosting the Service. With the example of the ‘MediaEmpty’ Service State Reason, the Current Condition table entry may contain details such as which tray is empty, the condition severity and when the condition occurred. When the condition is cleared the entry is moved to the Condition History table and the time the condition was cleared is added to the entry. Note however that there is not a one to one mapping between entries in the Condition Table and Service State Reasons entries. One condition may result in zero or more Service State Reasons and more than one service may be affected by a single condition.

While Service State and Service State Reason are intended for consumption by automata and can be localized for End User presentation by client software, State Messages is an optional status element that

is intended to be presented for direct End User consumption. State Messages is derived from IPP printer-state-message (RFC2911, para 4,4,13 [RFC2911]) and is a Server side localized set of strings that summarize the state of the Service. This element is used to provide information on the state of a Service in one or more message strings. The content of the string(s) is vendor specific and is intended to augment the information contained in Service State and Service State Reasons.

5 Imaging Job Model

The general Imaging Job model includes the Job Status, Job Ticket and Job Receipt Element groups and the Documents complex Element, as shown in Figure 85. The individual Services each have their own <service>Job model, paralleling this general Imaging Job model.

The state of the Job is described in the Job Status Element. The Job Ticket and the Job Receipt contain descriptive information about the Job and about the Job and Document processing instructions. The Job Ticket and the Job Receipt have the same structure, but the Element values in the Job Ticket are the information provided by the user in the CreateJob operation while the Element values in the Job Receipt reflect what was actually used in processing the Job.

Jobs can contain zero or more Documents. (During Job creation, it is possible that temporarily there are zero Documents.) Some Services do not expose the Document object. Therefore the Imaging Job model includes the optional Documents Element, which in turn may include zero or more Document Elements. It is possible to override the Job level Document Processing instructions on a Document by Document basis by supplying Document Processing Elements in Document Tickets for the affected Documents. The Document model (described in Chapter 6) mirrors the Job model, consisting of Document Status, Document Ticket and Document Receipt. The Document Ticket like the Job Ticket represents the End User's intent while the Document Receipt like the Job Receipt represents what the Service actually did.

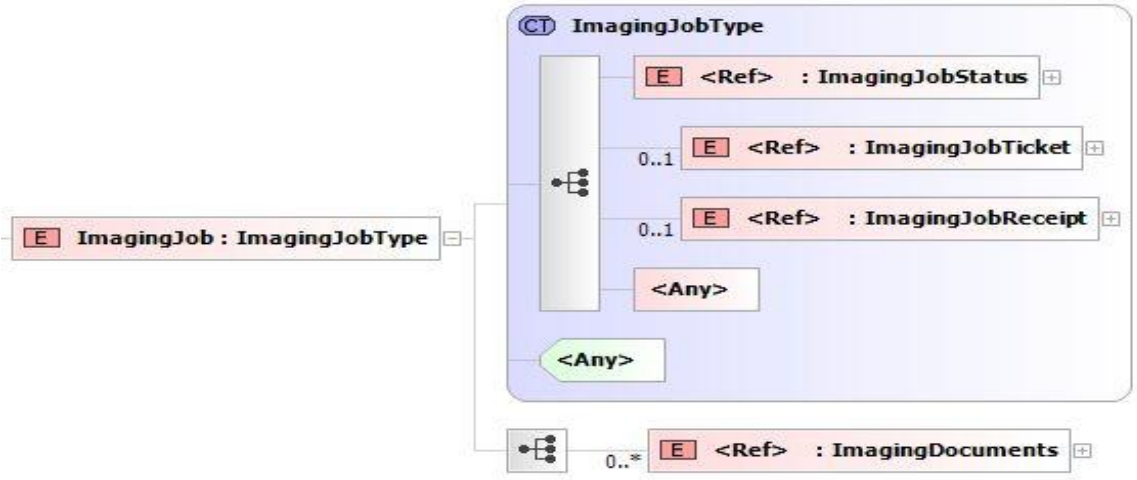

Figure 85 Imaging Job

5.1 Job Status

Job Status Elements provide state information for the Job. The <service> Job Status Elements typically include Elements from the generic Imaging Job Status as well as Service-specific Elements. Imaging Job Status is modeled in Figure 86, Figure 87 and Figure 88. The Elements are described in Table 58, Table 59 and Table 60. The values of these Elements are maintained by automata and cannot be set directly by a User or Administrator, although Element values can be modified indirectly through an operation. For example, the CancelJob operation on a Job may result in the change in values of the State and State Reasons Elements.

Some Service-specific Job Status Elements are represented in Figure 88 and listed in Table 60 All Services include a <service>Job Counters complex Element; the constituent counters are discussed in Chapter 8. Most include an Images Completed or Impressions Completed Element. Some Services include substantially more Elements in Job Status.

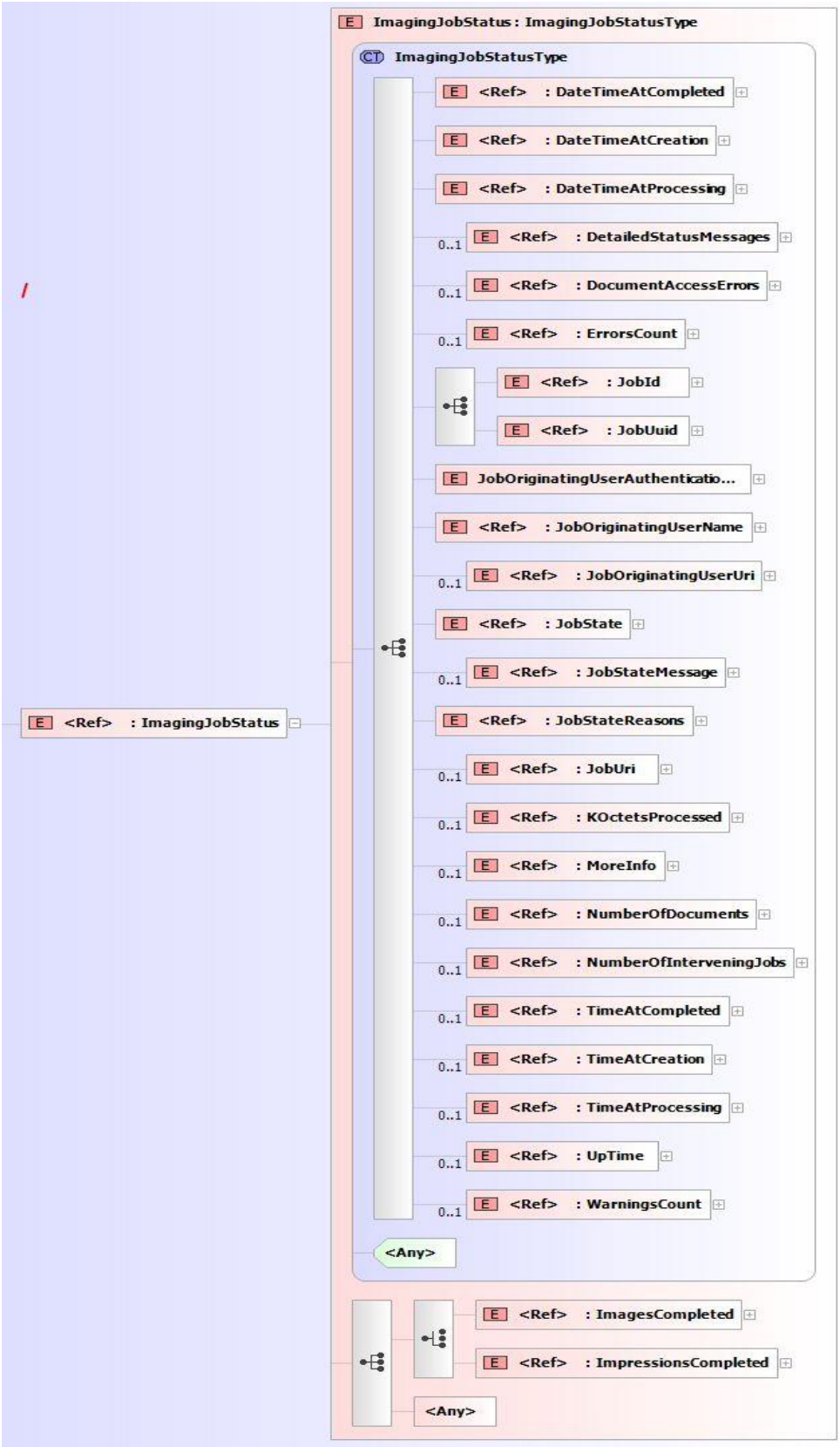

Figure 86 Imaging Job Status

Table 58 Job Status Elements

Element	DataType	Description or Keyword	Reference
DateTimeAtCompleted	dateTime	date and time at which the Job object processing was completed (or was canceled or aborted	[RFC2911] para. 4.3.14.7
DateTimeAtCreation	dateTime	date and time at which the Job object was created.	[RFC2911] para. 4.3.14.5
DateTimeAtProcessing	dateTime	date and time at which the Job object first began processing	[RFC2911] para. 4.3.14.6
DetailedStatusMessages	list of string	additional detailed and technical information about the Job. The Service MAY localize the message(s), since they are intended for use by the system administrator or other experienced technical persons. Localization might obscure the technical meaning of such messages. Clients MUST NOT attempt to parse the value of this Element.	[PWG5100.11] para. 10.8
DocumentAccessErrors	list of string	additional information about each Document access error for this Job encountered by the Service after it attempted to access or store the Image data at the locations supplied in the Job Ticket.	[RFC2911] para. 4.3.11
ErrorsCount	int	number of errors encountered during processing (e.g., for the Scan Service, while scanning the Hardcopy Document and generating and storing the Digital Document.)	[PWG5100.7] para. 5.1.1
JobId	int	uniquely identifies the Job within the Service. The Service is responsible for implementing a Job numbering scheme that will not allow two different Jobs to coexist with the same JobId.	[RFC2911] para. 4.3.2
JobUuid	anyUri	An urn::uuid unique URI value [RFC4122]	[RFC4122]
JobOriginatingUserAuthenticationInfo	complex	See Table 59.	[RFC2617], [RFC4556], [RFC2911]
JobOriginatingUserName	string	Service sets this to the most authenticated printable name that it can obtain (example: “John Doe”, \authDomain\John Doe”)	[RFC2911] para. 4.3.6
JobOriginatingUri	anyUri	the URL used in processing the job. For example this element would be populated with the source phone number for a FaxOut job.	See 5.1.1
JobState	keyword	current state of Job. The state values MUST NOT be extended by an implementation. From RFC2911, *JobStateWKV*	[RFC2911] para. 4.3.7
JobStateMessages	string	information about the Job State and StateReasons in human readable text. If the Service supports this Element, it MUST be able to generate the messages in any of the natural languages supported by the Service.	[RFC2911] para. 4.3.6
JobStateReasons	list of keywords	additional detail about the Job state. The typical keyword values are listed below. Values specific to a service are identified in the specification for that service. *JobStateReason*	para. 4.3.8 of [RFC2911] and para. 4.5.1.3 of [WS-Scan].
JobUri	string	globally unique identifier for a Job.	[RFC2911] para. 4.3.1
KOctetsProcessed	int	total number of octets processed at that time, in integral units of 1024 octets	[RFC2911] para. 4.3.18.1
MoreInfo	string	URI used to obtain information intended for End User consumption about this specific Job.	[RFC2911] para. 4.3.4
NumberOfDocuments	int	number of Documents in this Job.	[RFC2911] para. 4.3.12
NumberOfInterveningJobs	int	number of Jobs that are "ahead" of this Job, assuming the current scheduled order.	[RFC2911] para. 4.3.15
TimeAtCompleted	int	time at which the Job completed in “UpTime” seconds.	[RFC2911] para. 4.3.14.3
TimeAtCreation	int	time at which the Job was created in “UpTime” seconds.	[RFC2911] para. 4.3.14.1
TimeAtProcessing	int	time at which the Job first began processing in “UpTime” seconds.	[RFC2911] para. 4.3.14.2
UpTime	int	time duration (in seconds) that the service has been up and running	[RFC2911] para. 4.3.14.4

Element	DataType	Description or Keyword	Reference
WarningsCount	int	total number of warnings that a Service has generated while processing and delivering the Job's Document(s)	[PWG5100.4] para. 6.1
ImagesCompleted	int	Progress measure for Job in terms of output. Output may be Images for Hardcopy Document producers and/or Impressions for Hardcopy Document consumers. See PWG Counter Spec [PWG5106.1]	[PWG5106.1] para. 5.2.1
ImpressionsCompleted	int		[RFC2911] para. 4.3.18.2
Any	various	an extension point for JobStatus	

5.1.1 JobOriginatingUri

This Job Status element is the URL used in processing the job. For example this element would be populated with the source phone number for a FaxOut job. The abstract data type for this element is string.

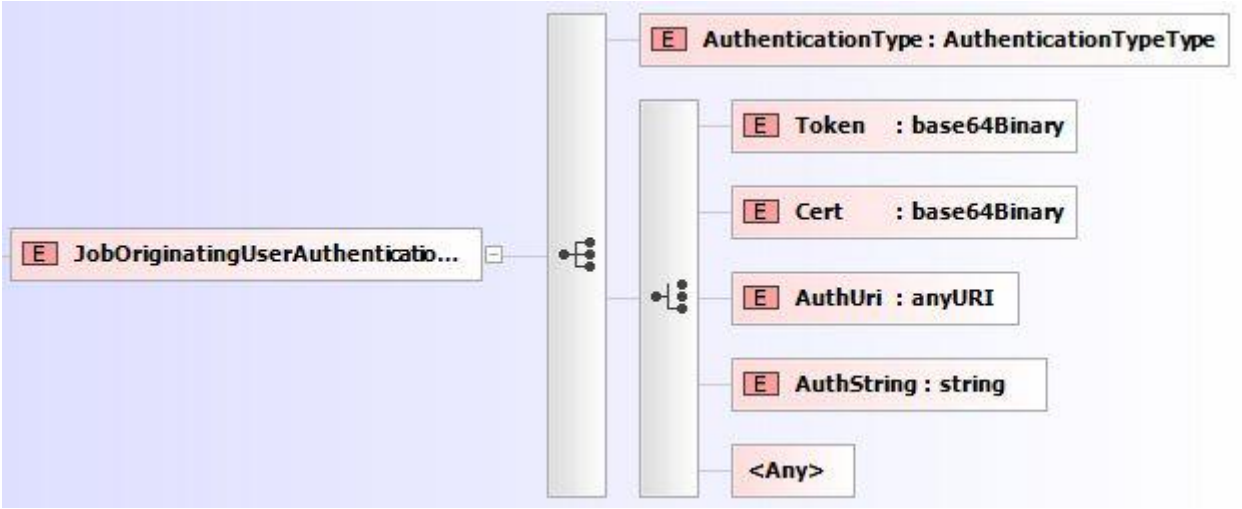

Figure 87 Job Originating User Authentication Info

Table 59 Job Originating User Authentication Info Elements

Element	DataType	Description	Reference
AuthenticationType	keyword	Type of authentication used: Basic, Digest: data in Token, Certificate: data in Cert, Kerberos: data in Token, Token: data in Token, RequestingUserName: data in AuthString, None: no associated data Any vendor extended keyword: any of the fields below	 RFC2617, e.g., RFC4158, RFC4556, Vendor specific content RFC2911 No reference Vendor specific content
Token	base64binary	Contains a transport safe encoded token	RFC4648
Cert	base64binary	Contains a transport safe encoded certificate	RFC4648
AuthUri	anyUri	A sting with a URI syntax	RFC2396
AuthString	string	A string	
Any	various	Vendor specific type other than one of the above	

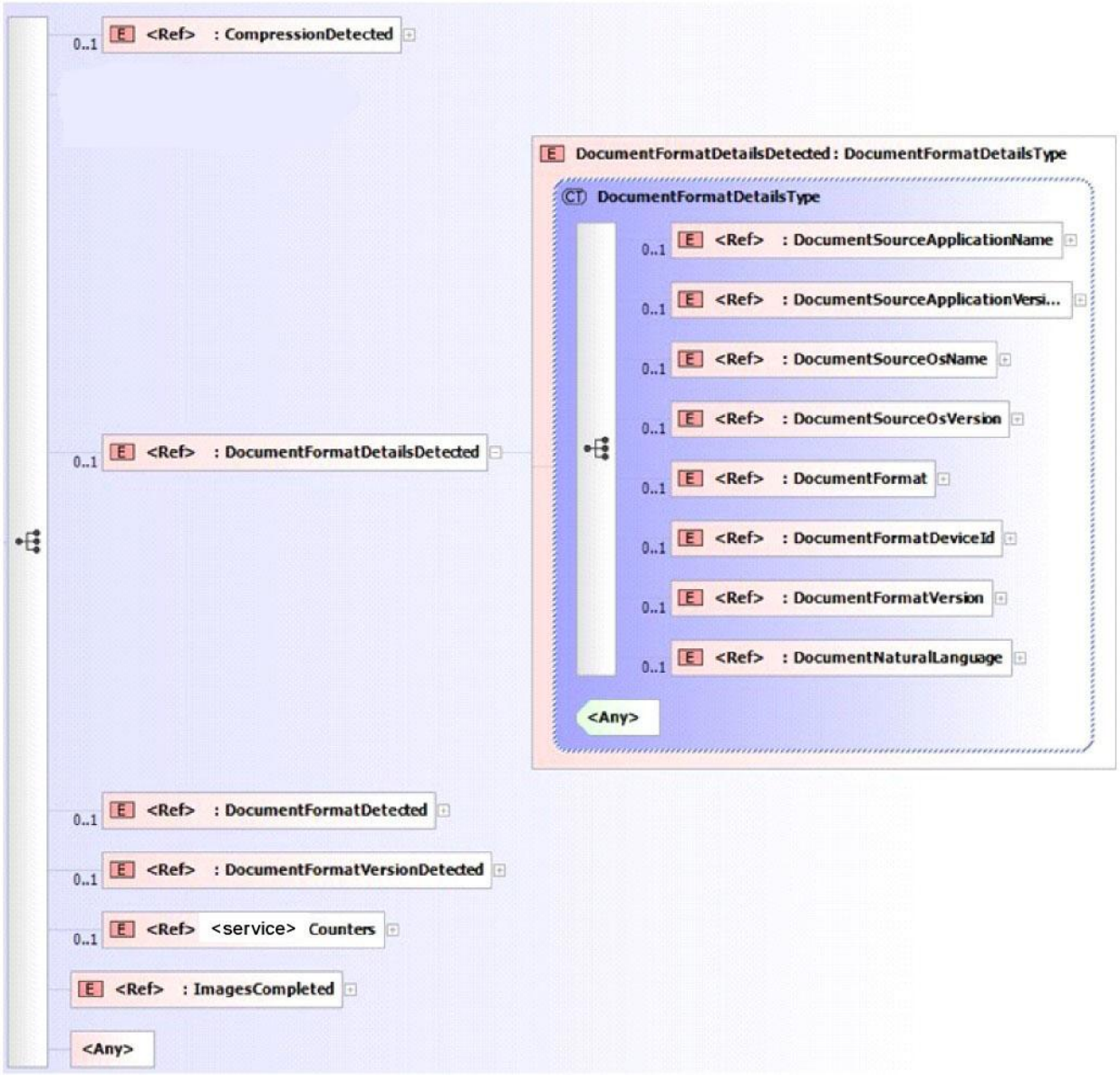

Figure 88 Representative Service-Specific Job Status Elements

Table 60 Service-Specific Job Status Elements

Element	DataType	Description	Reference
CompressionDetected	keyword	Compression type actually used on the Document	[PWG5100.5] para9.1.12
DocumentFormatDetailsDetected	complex	Generated by the service to indicate the actual document format details of the Document object	[PWG5100.5] para9.1.14
DocumentSourceApplicationName	string	name of the application that created the document, without its version number	[PWG5100.5] para9.1.13
DocumentSourceApplicationVersion	string	version of the application that created the document, without its name	[PWG5100.5] para9.1.13
DocumentSourceOsName	string	name of the operating system, without version number, on which the document was generated	[PWG5100.5] para9.1.13
DocumentSourceOsVersion	string	version of the operating system, without its name, on which the document was generated	[PWG5100.5] para9.1.13
DocumentFormat	keyword	Document format (i.e., PDL) for the Document	[RFC2911] para3.2.1.1 [PWG5100.5] para9.1.12
DocumentFormatDeviceId	string	Identifies the type of device for which the document was formatted, including manufacturer and model, following the IEEE 1284-2000 Device ID string	[PWG5100.5] para9.1.13
DocumentFormatVersion	string	the level or version of the DocumentFormat.	[PWG5100.5] para9.1.16
DocumentNaturalLanguage	keyword	Identifies the primary Natural Language of the Document.	[RFC2911] para3.2.1.1 [PWG5100.5] para9.1.22
DocumentFormatDetected	keyword	service sets this to the actual DocumentFormat detected when auto-sensing	[PWG5100.5] para9.1.15
DocumentFormatVersionDetected	string	service sets this to the actual DocumentFormat level detected when auto-sensing	[PWG5100.5] para9.1.17
<service>JobCounters	complex	Counters appropriate to the Service See Chapter 8	[PWG5106.1]
Images (or Impressions)Completed	int	Number of images (or impressions) processed by the service	[PWG5106.1]
Any	various	Extension point for <service>JobStatus	

5.2 Job Ticket

The generic Imaging Job Ticket (Figure 89) contains description and processing Elements provided by the client in the CreateJob request. This information is used by the Service during the processing of a Job and is made available to Clients through the GetJobElements operation. A subset of this information is made available through the GetActiveJobs and GetJobsHistory operations.

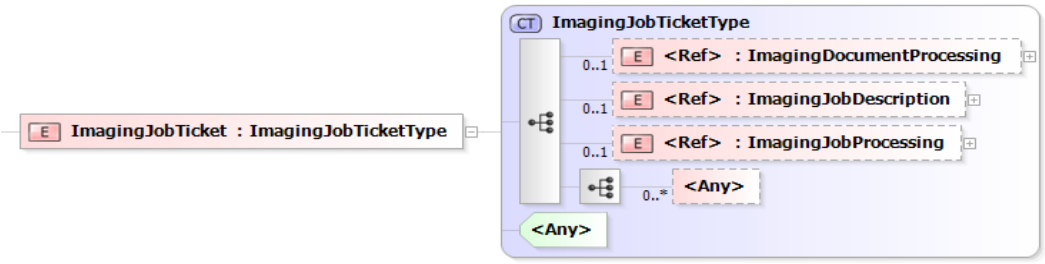

Figure 89 Imaging Job Ticket

The general Imaging Job Ticket indicates the Elements that may be included in a Job Ticket. The Service-specific models may contain additional Elements. A Job Ticket supplied with a CreateJob request need not contain all permitted Elements. Elements that are necessary but not supplied with the CreateJob

request are supplied the by the Default<service>Job Ticket that is part of the Service model (see Service Defaults, section 4.5.) The Default Job Ticket may contain all of the same Elements as are defined for the Job Ticket. However, the Element values within a Job Ticket are specific and single-valued.

5.2.1 Job Ticket Document Processing

The Document Processing complex Element provides the Document processing instructions that have been requested by the End User at the Job level. Each Element has a Must Honor Element to indicate whether Documents within the Job must be processed according to what user has requested or whether a “best effort” implementation is allowed. Note that the Document Processing Elements contained in the JobTicket follow the same general structure as Service Capabilities DocumentProcessing, but the Datatypes and values reflect specific instances reflecting the User’s intent.

Document Processing consists of two sequences:

- Elements inherited from the Imaging Service super class (just Number Up and Presentation Direction Number Up, as shown in Figure 90 and described in Table 61.
- Service specific Elements (a much larger set) that may be common to two or more Services. Generally, these elements are common among image-outputting Services such as Scan, FaxOut and EmailOut (Figure 91, Figure 92 and Figure 93 and Table 62), and among impression-outputting Services such as Print, FaxIn and EmailIn (Figure 94, Figure 95 and Figure 96 and Table 64). Some Services may have elements from both image and impression outputting groups.

Although the structure and names of constituent elements are generally the same as shown in the Service Model description (Chapter 4), the data types and descriptions of these Elements in a Job Ticket are as described in this section.

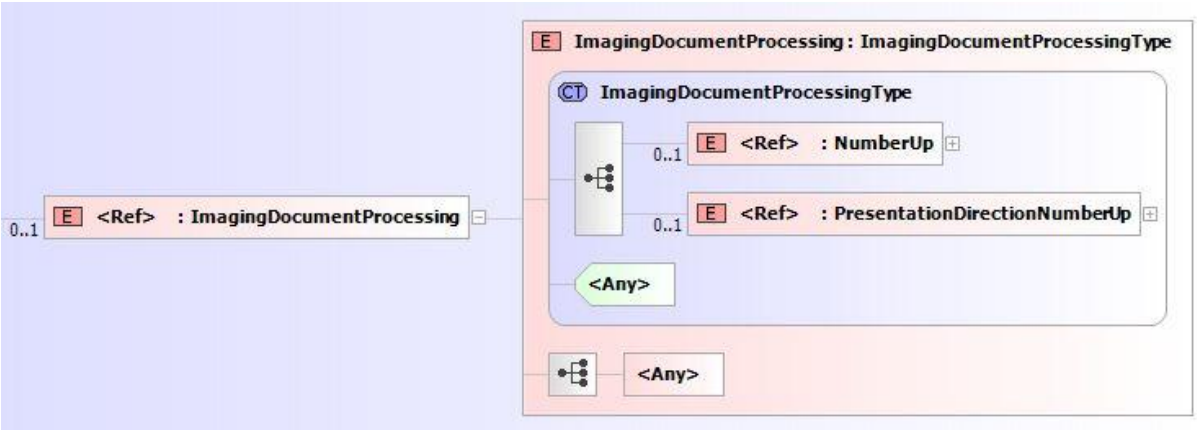

Figure 90 JobTicket Superclass DocumentProcessing

Table 61 JobTicket Superclass DocumentProcessing Elements

Element	DataType	Description or *Keyword Group*	Reference
NumberUp	fint	number of input “pages” that are ro be included in a single output “impression; or the number of input “images” that areto be included in the output image.	[RFC2911] para 4.2.9
PresentationDirectionNumberUp	keyword	placement order of the Images from the input Scan Regions or pdl pages on to the output Image or impression. Associated with the “NumberUp” Element *PresentationDirectionNumberUpWKV*	[PWG5100.3] para 3.17
Any	various	Extension point for ImagingDocumentProcessingCapabilities	

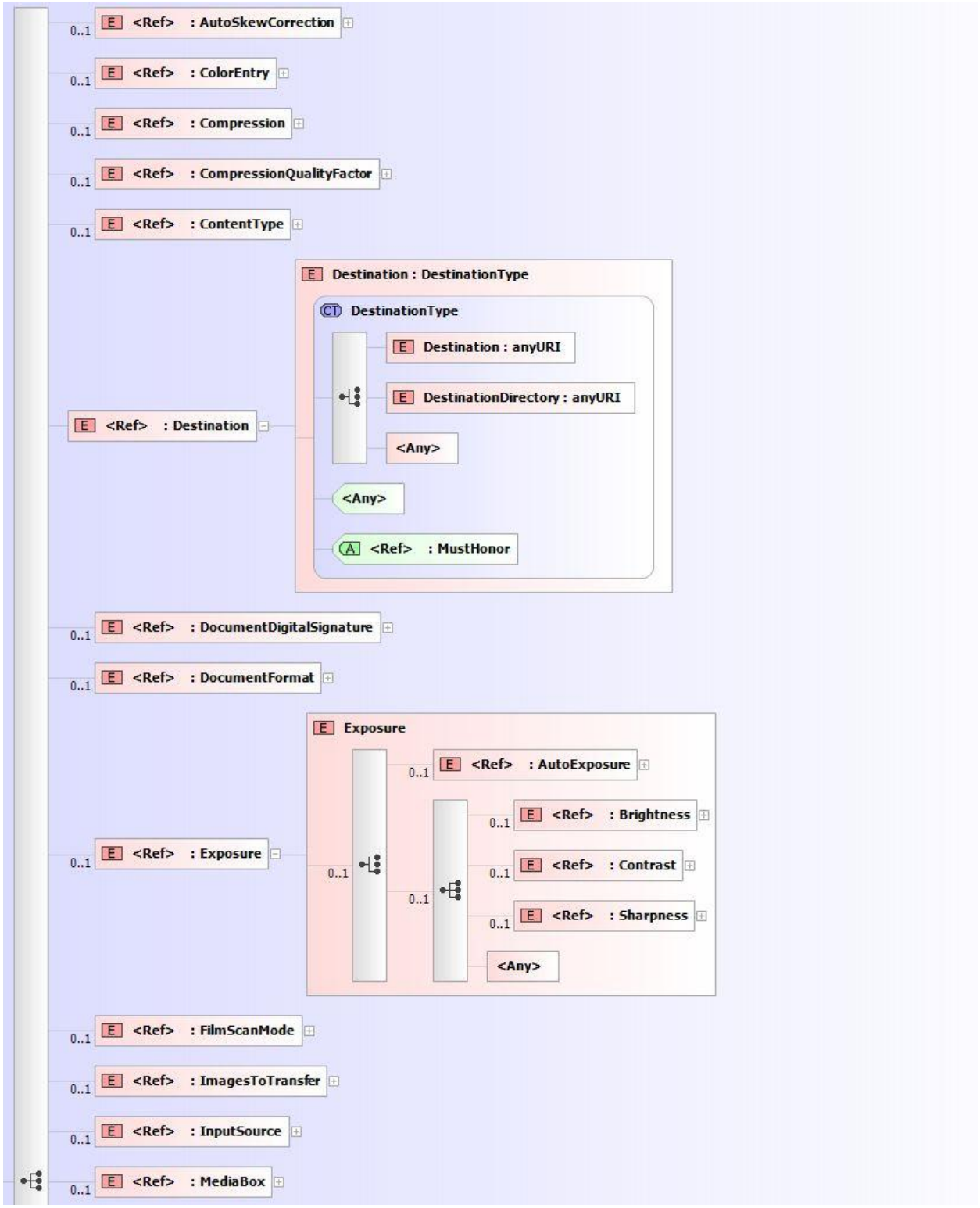

Figure 91 JobTicket DocumentProcessing - Image Output Services Sheet 1

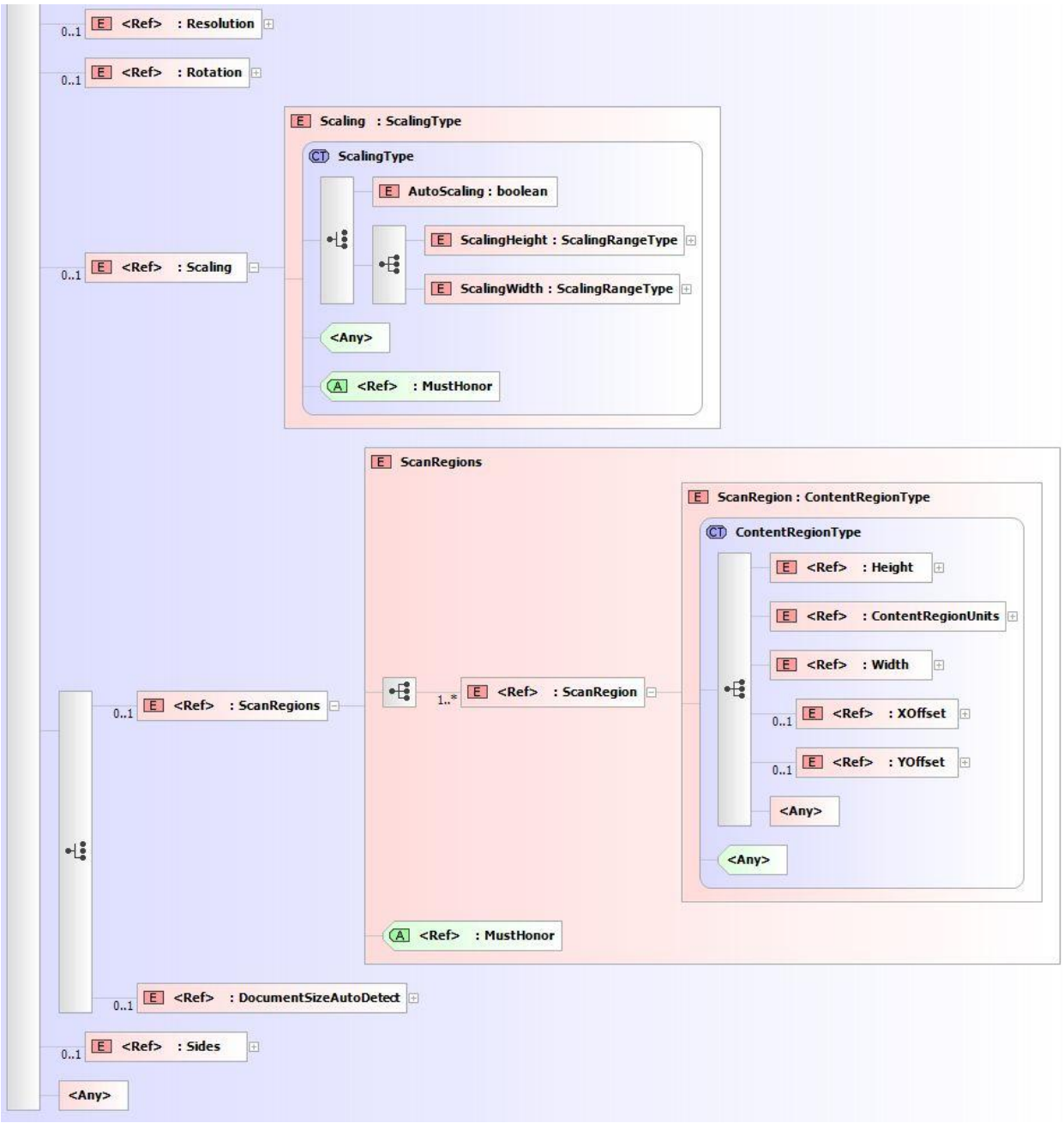

Figure 92 JobTicket DocumentProcessing - Image Output Services Sheet 2

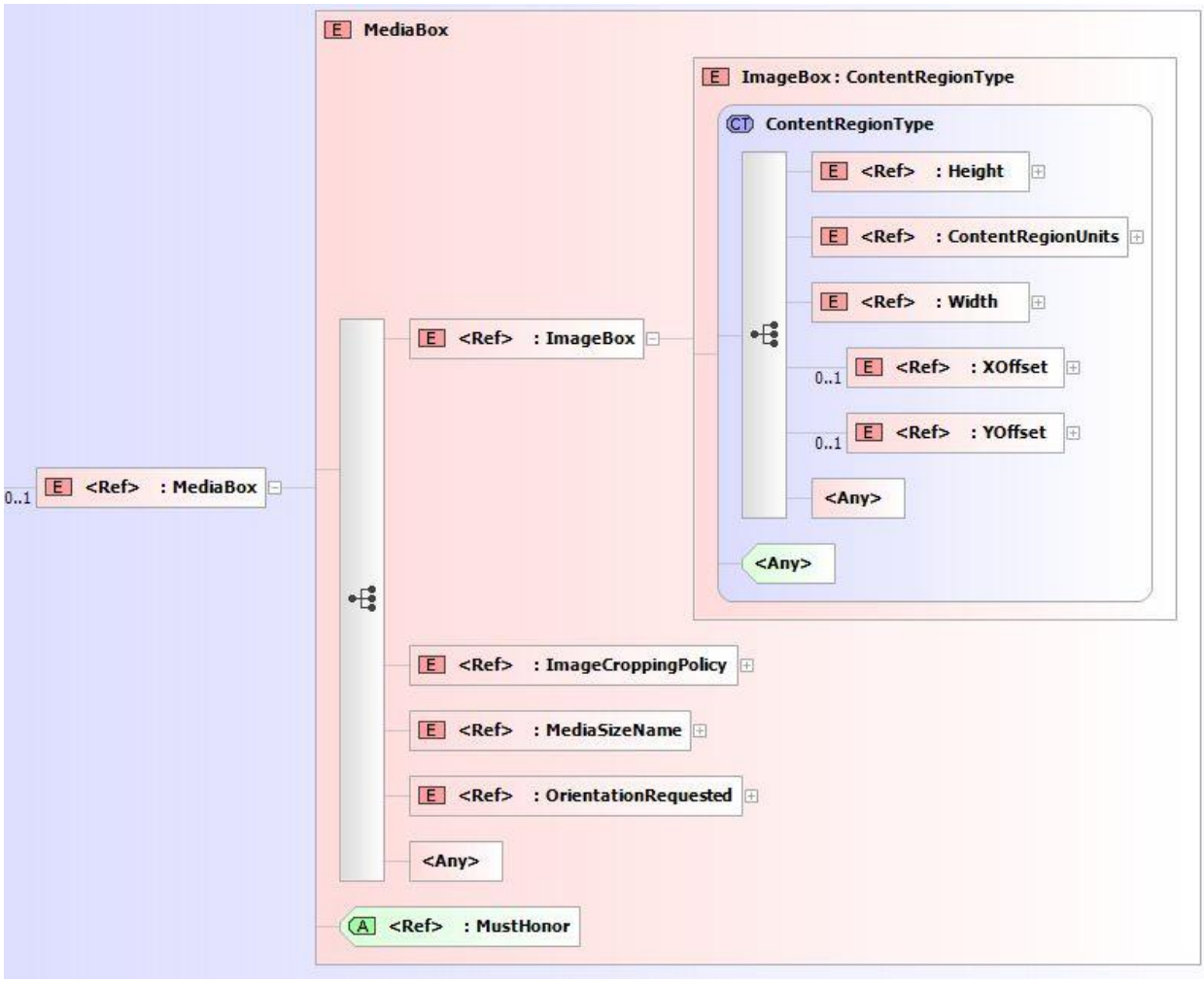

Figure 93 JobTicket DocumentProcessing - Image Output Services MediaBox

Table 62 Service-Specific Job Ticket Document Processing Elements - Image Output Services

Element	DataType	Description or *Keyword*	Reference
AutoSkewCorrection	boolean	detection and correction of small skew orientation error from the media loading from the ADF or placement on the platen by the user is to be applied.	[PWG5108.2] para 8.1.3.1.3
ColorEntry	keyword	color processing mode. Each keyword describes a color encoding, color space, bit depth and samples per pixel combination. See Table 63 for keywords Vendors may add additional keywords. *ColorEntryWKV*	[PWG5108.2] para 8.1.3.1.4
Compression	keyword	compression algorithm used on the Document Data, if any. *CompressionWKV*	[RFC2911] para 4.4.32
CompressionQualityFactor	int	normalized integer value used by JPEG compression to determine the amount of acceptable image loss. JPEG compression can be lossy, some amount of data is lost (not reproducible) or lossless. The higher the requested compression factor the smaller the resulting file size. The value is normalized as an integer between 0 and 100.	[PWG5108.2] para 8.1.3.1.6
ContentType	keyword	main characteristics of the original Document. It is used as a hint to the Scan Service on how it should handle the scanning of the Hardcopy Document. *ContentTypeWKV*	[PWG5108.2] para 8.1.3.1.7
Destination	complex	destination to which output of the Job s to be delivered. Either a directory in which the Digital Document is stored or the URI to the Digital Document file.	[PWG5108.2] para 8.1.3.1.8
Destination	anyURI		[PWG5108.2] para 8.1.3.1.8.1
DestinationDirectory	anyURI	used when the output of the Job is a set of Digital Document files stored in a directory.	[PWG5108.2] para 8.1.3.1.8.2
Any	various	Extension point for Destination	
DocumentDigitalSignature	keyword	type of digital signature, if any, used in the creation of the Digital Document. *DocumentDigitalSignatureWKV, KeywordNsExtensionPattern*	[PWG5100.7] para 3.2.3 [PWG5100.5] para 9.1.11
DocumentFormat	keyword	format used to save Digital Documents. This is given as an enumeration value of MIME type which is pwg:DocumentFormat type. *DocumentFormatWKV*	[RFC2911] para 3.2.1.1 and [PWG5100.5] para 9.1.12
DocumentSizeAutoDetect	boolean	area within the input Hardcopy Document's media sheet side boundaries to scan are automatically determined by the device in an implementation specific manner. This Element is mutually exclusive with Scan Region.	[PWG5108.2] para 8.1.3.1.20
Exposure	complex	AutoExposure Element or a sequence of four Elements: Brightness, Contrast, Sharpness, and Any.	[PWG5108.2] para 8.1.3.1.11
AutoExposure	boolean	automatic background reduction will be performed on the input Document.	[PWG5108.2] para 8.1.3.1.11.1
Brightness	int	relative amount to enhance or reduce the brightness of the scanned Image. Negative values will darken the Image and positive values will lighten the Image. The value is normalized as an integer between -100 and 100.	[PWG5108.2] para 8.1.3.1.11.2
Contrast	int	relative amount to enhance or reduce the contrast of the scanned Image Negative values will reduce the apparent difference between light and dark pixels in the Image. Positive values will increase the apparent difference between light and dark pixels in the Image. The value is normalized as an integer between -100 and 100.	[PWG5108.2] para 8.1.3.1.11.3
Sharpness	int	relative amount to enhance or reduce object edges within a scanned Image. Positive values enhance object edges and negative values reduce object edges. The value is normalized as an integer between -100 and 100.	[PWG5108.2] para 8.1.3.1.11.4
Any		extension point forExposure	

Element	DataType	Description or *Keyword*	Reference
FilmScanMode	keyword	exposure type of the film to be scanned. The Element is valid only if the InputSource Element is set to a value of 'FilmReader'. *FilmScanModeWKV, KeywordNsExtensionPattern*	[PWG5108.2] para 8.1.3.1.12
ImagesToTransfer	int	number of Images to scan from the Hardcopy Document.	[PWG5108.2] para 8.1.3.1.13
InputSource	keyword	source of the scanned Document. *InputSourceWKV*	[PWG5108.2] para 8.1.3.1.14
MediaBox	complex	Applicable only when the Document format used to store the Digital Document has a similar construct that controls the placement and bounding of the scanned Image on the Document format specific page Image. Prior to applying all other processing Elements are assumed to have been applied.	[PWG5108.2] para 6.5.3
ImageBox	complex	region where the Image from the scan of the Scan Region will be placed.	[PWG5108.2] para 6.5.3
Height	int	height of the image box in RegionUnits.	[PWG5108.2] para 7.1.2.1.19.1
Width	int	width of the image box in RegionUnits.	[PWG5108.2] para 7.1.2.1.19.2
X-Offset	int	offset, positive or negative, on the X axis from the origin of the MediaBox in RegionUnits.	[PWG5108.2] para 7.1.2.1.19.3
Y-Offset	int	offset, positive or negative, on the Y axis from the origin of the MediaBox in RegionUnits	[PWG5108.2] para 7.1.2.1.19.4
Any	various	extension point for ImageBox	
RegionUnits	keyword	unit used for specifying all dimensions defining the region. *RegionUnitWKV*	[PWG5108.2] para 7.1.2.1.19.5
ImageCroppingPolicy	keyword	controls handling of mismatches in the size of the Image and the intersection of the ImageBox and MediaBox where the Image will be placed. *ImageCroppingPolicyWKV, KeywordNsExtensionPattern*	[PWG5108.2] para 8.1.3.1.15.2
MediaSizeName	keyword	identifies the size of the canvas for an Image in the Digital Document expressed as a media size name. *MediaSizeNameType*	[PWG5101.1] para 5
OrientationRequested	keyword	identifies the orientation of the MediaBox *OrientationRequestedType*	[RFC2911] para 4.2.10
Any	various	Extension point for MediaBox	
Resolution	complex	resolution in the Feed and Cross Feed directions at which to impress or capture the Image. The Resolution data structure does not limit the values. However, individual devices may limit the range of options to a predefined list or range. This information will be available as part of the Service Capabilities provided as a result of a Get,serevice>Elements query. Resolution is specified in pixels per inch or pixels per centimeter. Constituent elements of Resolution are described in Table 64.	RFC2911] para 4.2.12
Rotation	keyword	amount each Image of a scanned Document is to be rotated. Specified in limted set of values in degrees clockwise. Standard values are 90, 180, 270. Vendors may add more. *RotationWKV*	[PWG5108.2] para 8.1.3.1.17
Scaling	complex	scaling to be applied to the scanned Image. Isomorphic (i.e., the aspect ratio of the Image does not change) is accomplished by specifying the same values for ScalingWidth and ScalingHeight. A value of '100' specifies that no adjustments are made to the scanned Image. Magnification is expressed in 1 percent increments. Values below '100' reduce the magnification and values above increase magnification.	[PWG5108.2] para 8.1.3.1.18
ScalingHeight	int	scaling in the fast scan direction expressed in 1 percent increments. Values below '100' reduce the magnification and values above increase magnification.	[PWG5108.2] para 8.1.3.1.18.1
ScalingWidth	int	scaling in the slow scan direction expressed in 1 percent increments. Values below '100' reduce the magnification and values above increase magnification.	[PWG5108.2] para 8.1.3.1.18.2
AutoScaling	boolean	scaling of the image to be automatic, in an implementation specific manner.	[PWG5108.2] para 8.1.3.1.18.3

Element	DataType	Description or *Keyword*	Reference
ScanRegions	complex	areas within the input Hardcopy Document's media sheet side boundaries to scan. If Scan Region is not specified, the device should use 0 as the offsets and the width and height of the InputSize, if given. If Scan Region is not specified and InputSize is not specified or cannot be determined by the device, the implementation is up to the hardware vendor. Is mutually exclusive with DocumentSizeAutoDetect.	[PWG5108.2] para 8.1.3.1.19
ScanRegion	complex	See 2.4.1	[PWG5108.2] para 8.1.3.1.19.1
Height	int	height of the Scan Region in RegionUnits	[PWG5108.2] para 8.1.3.1.19.1.1
Width	int	width of the Scan Region in RegionUnits Width corresponds to the fast scan direction and Height corresponds to the slow scan direction.	[PWG5108.2] para 8.1.3.1.19.1.2
X-Offset	int	offset, positive or negative, on the X axis from the West Margin of the Scanner Subunit in RegionUnits	[PWG5108.2] para 8.1.3.1.19.1.3
Y-Offset	int	offset, positive or negative, on the Y axis from the West Margin of the Scanner Subunit in RegionUnits	[PWG5108.2] para 8.1.3.1.19.1.4
ContentRegionUnits	keyword	units used for specifying the region *RegionUnitWKV*	[PWG5108.2] para 8.1.3.1.19.1.5
Any	various	Extension point for ScanRegion	
DocumentSizeAutoDetect	boolean	True= document size automatically determined	
Sides	keyword	*SidesWKV*	
Any	various	extension point for JobTicketDocumentProcessing	

Note: the ColorEntry Element describes the color processing mode. Each keyword describes a color encoding, color space, bit depth and samples per pixel combination. Vendors may add additional keywords.

Table 63 Color Processing Elements

Keyword	Color Type	Color Encoding	Bit Depth	Samples per pixel
BlackandWhite1	Binary		1	1
Grayscale4	Gray		4	4
Grayscale8	Gray		8	8
Grayscale16	Gray		16	16
RGB24	color	RGB	24	8
RGB48	color	RGB	48	16
RGBa32	color	RGB	32	8
RGBa64	color	RGB	64	16
CMYK32	color	CMYK	32	8
CMYK64	color	CMYK	64	16

5.2.1.1 HeaderPrint

This Document Processing element specifies Indicates how a transmitted document header is added to the outbound transmission by a service (e.g., FaxOut EmailOut). The data type for this element is keyword. This element can be extended by adding new keywords. The corresponding Document Ticket Capabilities element is a list of keywords specifying the values supported by the implementation. The defined keywords are:

- 'Inside' – the header is added within document image for transmission,
- 'Outside' – The header is added outside the document image for transmission,
- 'NoPrint' – the header is not added to the document image for transmission.

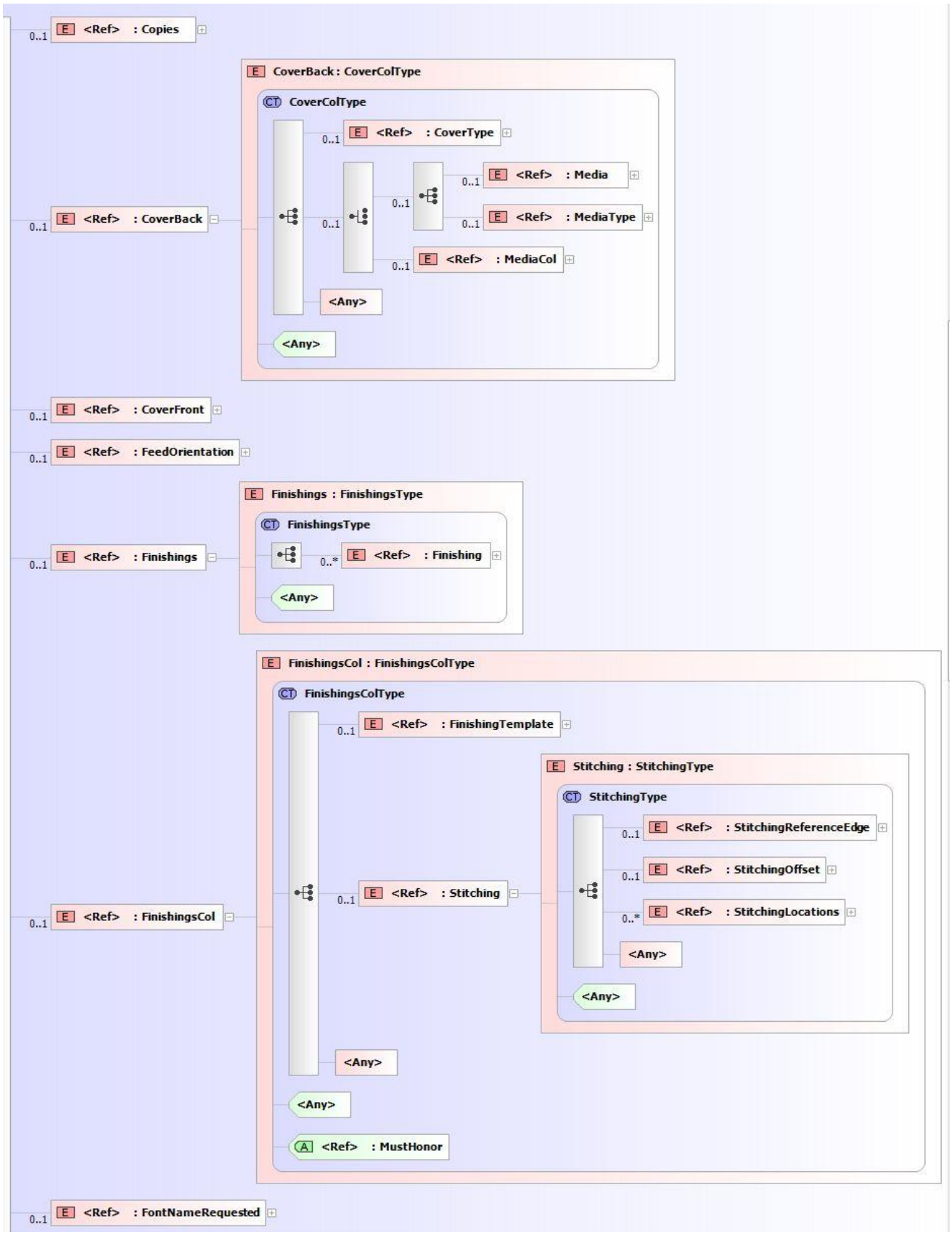

Figure 94 JobTicket DocumentProcessing - Impression Output Services Sheet 1

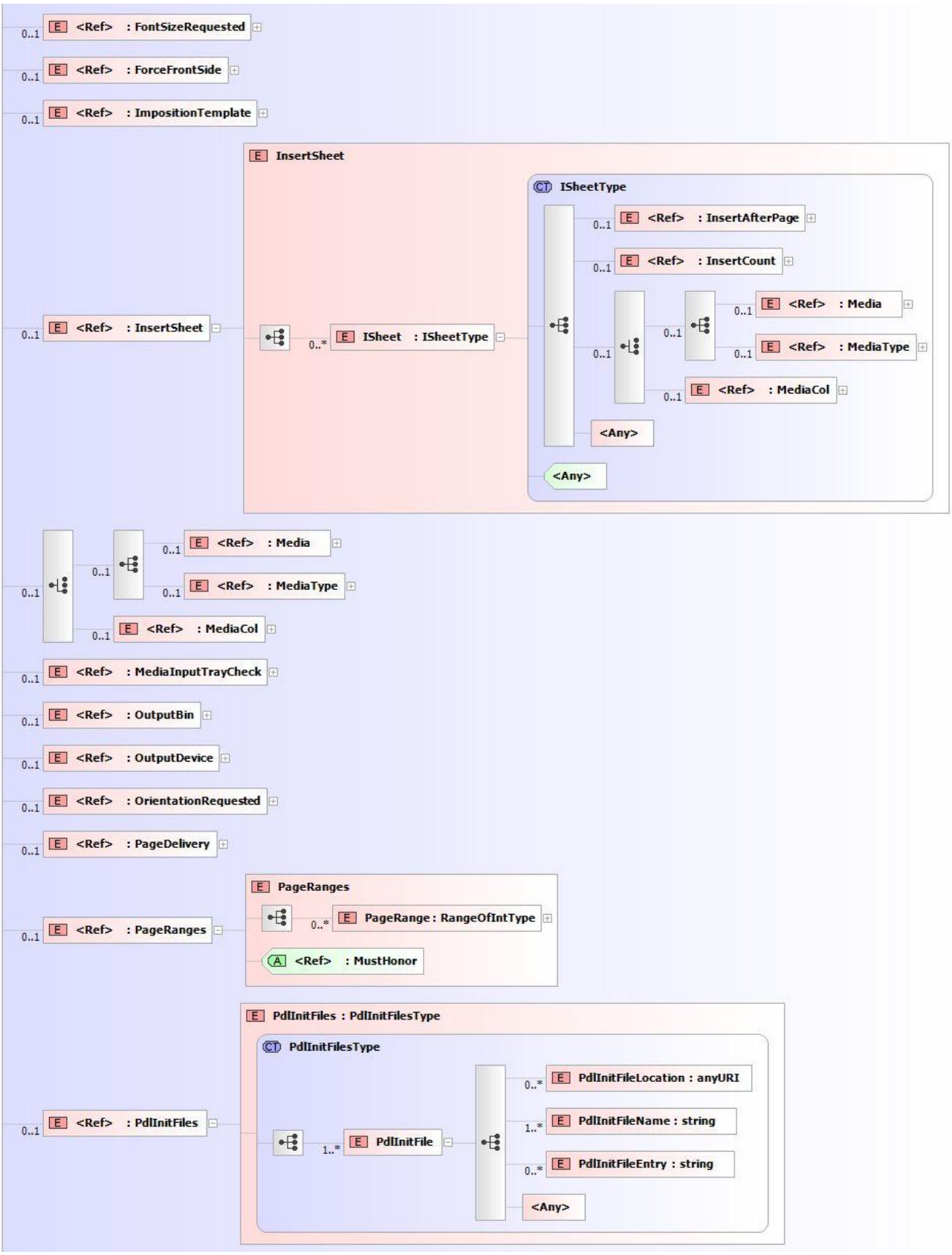

Figure 95 JobTicket DocumentProcessing – Impression Output Services Sheet 2

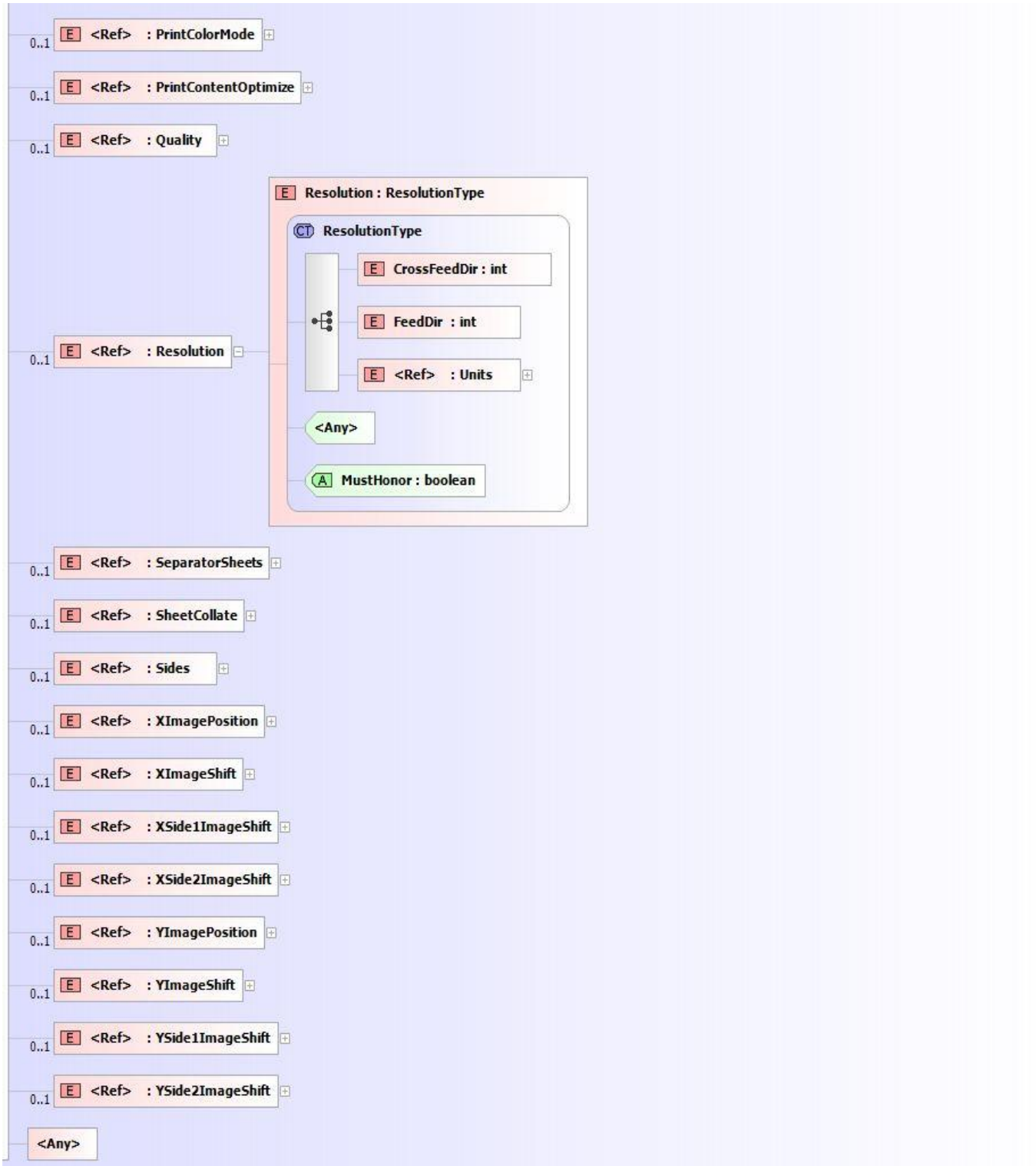

Figure 96 JobTicket DocumentProcessing – Impression Output Services Sheet 3

Table 64 Service-Specific Job Ticket Document Processing Elements - Impression Output Services

Element	DataType	Description or *Keyword*	Reference
Copies	int		[RFC2911] para 4.2.5
CoverBack	complex	Note that all complex Elements referring to a sheet of physical media upon which an impression could be made, are of the same structure as illustrated for CoverBack. This applies to CoverFront, ISheetType, SeparatorSheet, eyc.	[PWG5100.3] para 3.1
CoverType	keyword	*CoverTypeWKV*, KeywordNsExtensionPattern	[PWG5100.3] para 3.1.2
Media	keyword	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*	[RFC2911] para 4.2.11
MediaType	keyword	*MediaTypeWKV*	[PWG5100.3] para 3.13.2
MediaCol	complex	See Table 65	[PWG5100.3] para 3.13
Any	various	Extension point for CoverBack	[PWG5100.3] para 3.13.2
CoverFront	complex		[PWG5100.3] para 3.1
CoverType	keyword	*CoverTypeWKV,* KeywordNsExtensionPattern	[PWG5100.3] para 3.1.2
Media	keyword	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*	[RFC2911] para 4.2.11
MediaType	keyword	*MediaTypeWKV*	[PWG5100.3] para 3.13.2
MediaCol	complex	See Table 65	[PWG5100.3] para 3.13
Any	various	Extension point for CoverFront	
FeedOrientation	keyword	FeedOrientationWKV KeywordNsExtensionPattern	[PWG5100.11] para 7.1
FilmScanMode	keyword	exposure type of the film to be scanned. The Element is valid only if the InputSource Element is set to a value of 'FilmReader'. *FilmScanModeWKV, KeywordNsExtensionPattern*	[PWG5108.2] para 8.1.3.1.12
Finishings	keyword	FinishingsWKV, KeywordNsExtensionPattern	[RFC2911] para 4.2.6 [PWG5100.1] para 2
FinishingsCol	complex		[PWG5100.3] para 3.2
FinishingTemplate	string	A string specifying some particular finishing operation	[PWG5100.3] para 3.2.1
Stitching	complex		[PWG5100.3] para 3.2.2
StitchingReferenceEdge	keyword	Specifies the stitching reference edge of the output media StitchingReferenceEdgeWKV, KeywordNsExtensionPattern	[PWG5100.3] para 3.2.2.1
StitchingOffeset	int	The perpendicular distance from the reference edge to the stitching axis in hundredths of a millimeter.	[PWG5100.3] para 3.2.2.2
StitchingLocations	int	The distance along the stitching axis where a stitch will be placed in hundredths of a millimeter	[PWG5100.3] para 3.2.2.3
Any	various	Extension point for Stitching	
Any	various	Extension point for FinishingsCol	
FontNameRequested	string	Specifies the font name for a document format that does not have inherent font information	[PWG5100.11] para 7.2
FontSizeRequested	int	Specifies the font size for a document format that does not have inherent font information	[PWG5100.11] para 7.3

Element	DataType	Description or *Keyword*	Reference
ForceFrontSide	int	Forces the specified pages to be printed on the front side of a sheet of media.	[PWG5100.3] para 3.3
HeaderPrint	keyword	Indicates how a transmitted document header is added to the outbound transmission by the service (FaxOut EmailOut) *HeaderPrintWKV,* KeywordNsExtensionPattern	See 5.2.1.1
ImpositionTemplate	string	Specifies imposition method for laying out finished page images onto the surface of output media	[PWG5100.3] para 3.4
InsertSheet	complex	Specifies how Insert Sheets are to be inserted into the sequence of media sheets that are produced for each copy of the documents	[PWG5100.3] para 3.5
ISheet			
InsertAfterPage	int	Specifies the input page after which the Insert Sheet will be placed.	[PWG5100.3] para 3.5.1
InsertCount	int	Specifies the number of Insert Sheet to insert	[PWG5100.3] para 3.5.2
Media	keyword	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*	[RFC2911] para 4.2.11
MediaType	keyword	*MediaTypeWKV*	[PWG5100.3] para 3.13.2
MediaCol	complex	See Figure 97, and Table 65	[PWG5100.3] para 3.13
Any	various	Extension point for InsertSheet	
Media	keyword	*MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV*	[RFC2911] para 4.2.11
MediaType	keyword	*MediaTypeWKV*	[PWG5100.3] para 3.13.2
MediaCol	complex	Figure 97, and Table 65	[PWG5100.3] para 3.13
MediaInputTrayCheck	keyword	characteristics of the media in the identified input tray must match the characteristics of the media identified by the "media" or "media-col" element. *MediaInput TrayCheckWKV,* MediaNsExtensionPattern	[PWG5100.3] para 3.14
OrientationRequested	keyword	The desired orientation for printed pages for document formats that don't have a built-in orientation. *OrientationRequestedWKV*	[RFC2911] para 4.2.10
OutputBin	keyword	Specifies the output bin where the job is to be delivered. *OutputBinWKV*, KeywordNsExtensionPattern	[PWG5100.2] para 2.1 [PWG5100.5] para 8.1
OutputDevice	string	Specifies the device where the pages of of a Job/Document will be printed	[PWG5100.7] para 4.2.1 [PWG5100.5] para 8.1
PageDelivery	keyword	Indicates whether the pages of the job are to be delivered to the output bin or finisher in the same page order as the original document and face up or face down *PageDeliveryWKV*, KeywordNsExtensionPattern	[PWG5100.3] para 3.15
PageRanges	range of int	Specifies a range of pages in the document data to be output	[RFC2911] para 4.2.7
PagesPerSubset	int	Combines all of the Pages of all of the Documents into a single stream of -Pages. Then the Printer partitions that single stream into contiguous subsets of -Pages according to the list of integers	[PWG5100.4] para 5.3
PdlInitFile	complex	(Note: PdlInitFiles is an artifact of the XML encoding)	[PWG5100.11] para 7.10
PdlInitFileLocation	Uri	Absolute URI that specifies the path to the directory where the initialization file to be sent to the Printer's PDL interpreter will be found	[PWG5100.11] para 7.10.1.1
PdlInitFileName	string	specifies the name of the initialization file within the directory	[PWG5100.11] para 7.10.1.2

Element	DataType	Description or *Keyword*	Reference
PdlInitFileEntry	string	specifies an entry point within the init file at which the PDL interpreter starts.	[PWG5100.11] para 7.10.1.2
PrintContentOptimize	keyword	*PrintContentOptimizeWKV*, KeywordNsExtensionPattern	[PWG5100.7] para 4.2.2 [PWG5100.5] para 8.1
Quality	keyword	*PrintQualityWKV*, KeywordNsExtensionPattern	[RFC2911] para 4.2.13
Resolution	complex	resolution in the Feed and Cross Feed directions at which to impress or capture the Image. The Resolution data structure does not limit the values. However, individual devices may limit the range of options to a predefined list or range. This information will be available as part of the Service Capabilities provided as a result of a Get,serevice>Elements query. Resolution is specified in pixels per inch or pixels per centimeter.	RFC2911] para 4.2.12
CrossFeedDir	int	resolution in Units for the Cross Feed directions at which to capture the Image	RFC2911] para 4.1.15
FeedDir	int	resolution in Units for the Feed directions at which to capture the Image	RFC2911] para 4.1.15
Units	keyword	units used to express the resolution *UnitsWKV*	RFC2911] para 4.1.15
SeparatorSheets	complex	Specifies the separator sheets to be printed with the Document	[PWG5100.3] para 3.18
SpreaderSheetsType	keyword	Specifies the separator sheets type *SeparatorSheetsTypeWKV*, KeywordNsExtensionPattern	[PWG5100.3] para 3.18.1
Media	keyword	MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV	[RFC2911] para 4.2.11
MediaType	keyword	MediaTypeWKV	[PWG5100.3] para 3.13.2
MediaCol	complex	See Figure 97 and Table 65	[PWG5100.3] para 3.13
Any	various	Extension point for SeparatorSheets	
SheetCollate	keyword	*SheetCollateWKV*	[PWG5100.3] para 3.18.1
Sides	keyword	*SidesWKV*	
XImagePosition	keyword	*XImagePositionWKV*	[PWG5100.3] para 3.19.2
XImageShift	int	Causes the Finished-Page Image to be shifted in position with respect to the x-axis of the media	[PWG5100.3] para 3.19.3
XSide1ImageShift	int	Causes each Finished-Page Image that would be placed on the front side of a sheet to be shifted in position with respect to the x-axis of the media	[PWG5100.3] para 3.19.4
XSide2ImageShift	int	Causes each Finished-Page Image that would be placed on the backside of a sheet to be shifted in position with respect to the x-axis of the media	[PWG5100.3] para 3.19.5
YImagePosition	keyword	*YImagePositionWKV*	[PWG5100.3] para 3.19.6
YImageShift	int	Causes the Finished-Page Image to be shifted in position with respect to the y-axis of the media	[PWG5100.3] para 3.19.7
YSide1ImageShift	int	Causes each Finished-Page Image that would be placed on the front side of a sheet to be shifted in position with respect to the y-axis of the media	[PWG5100.3] para 3.19.8
YSide2ImageShift	int	Causes each Finished-Page Image that would be placed on the backside of a sheet to be shifted in position with respect to the y-axis of the media	[PWG5100.3] para 3.19.9
Any	various	extension point for JobTicketDocumentProcessing	

5.2.1.2 Media Collection (Specific)

The collection of media attributes, grouped in the Media Col complex Element, appears many places where media is to be defined, or capabilities of handling media are identified. The Media Col *capabilities* structure was shown in Figure 68 with the constituent *capabilities* elements listed in Table 45. The Media Col structure when referring to a specific item of media (a medium) is represented in Figure 97, with the Elements listed in Table 65.

Media Col contains many member Elements. Therefore, unlike for other Job or Document Elements, Services should not return Media Col values in response to a Get<service>JobElements or Get<service>JobElements request. An explicit request for the specific Media Col elements is required.

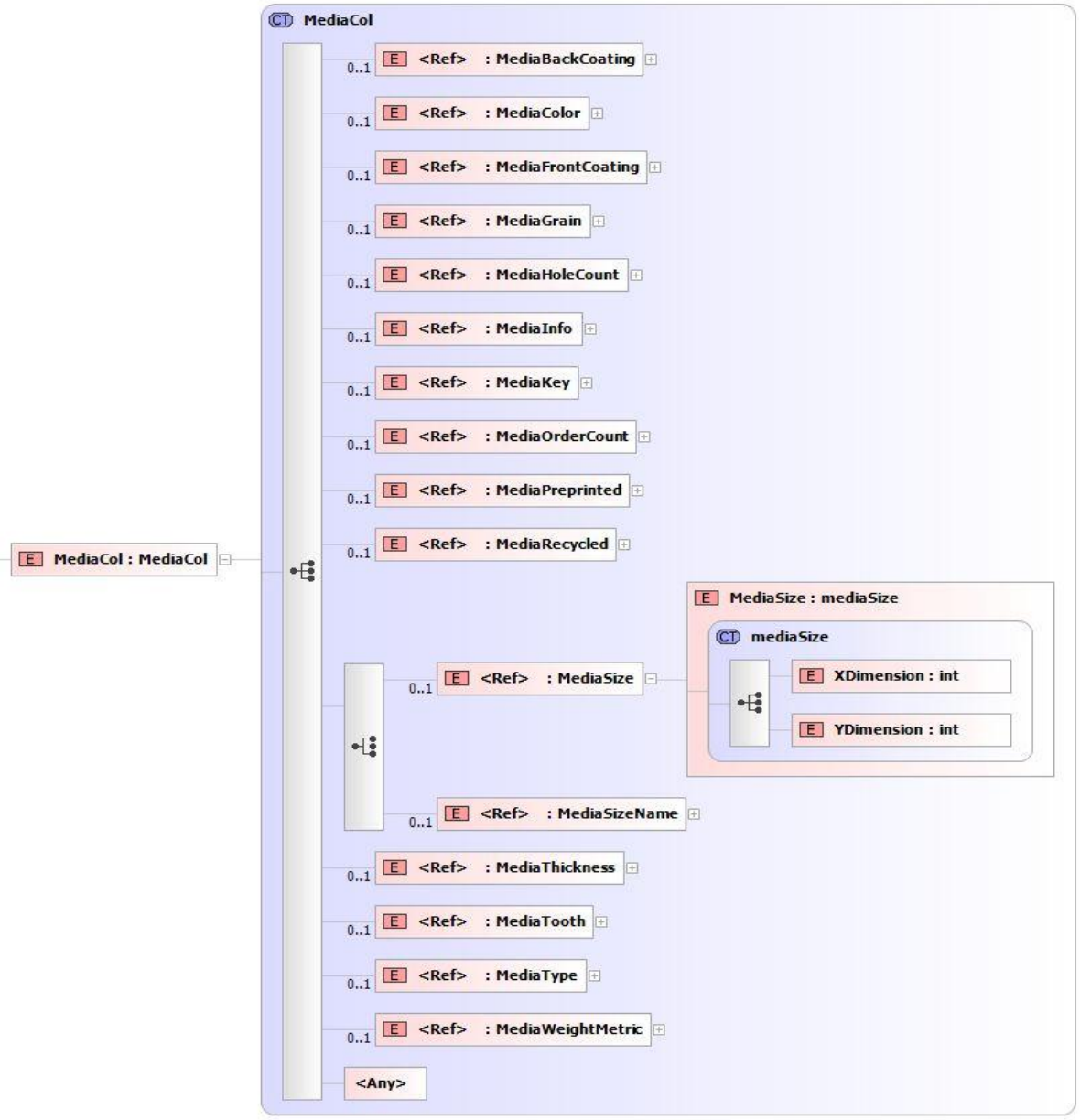

Figure 97 Media Col Structure Specific

Table 65 Media Collection Elements in Job Ticket

Element	DataType	Description or *Keyword*	Reference [PWG5100.3]
MediaCol			
MediaBackCoating	keyword	pre-process coatings applied to the media *MediaCoatingWKV*, MediaColorExtensionPattern	para 3.13.10
MediaColor	keyword	color of the media *MediaColorWKV*, MediaColorExtensionPattern	para 3.13.4
MediaFrontCoating	keyword	pre-process coatings applied to the media *MediaCoatingWKV*, MediaNsExtensionPattern	para 3.13.10
MediaGrain	keyword	grain of the media (affects curling and folding) *MediaGrainWKV*, MediaNsExtensionPattern	[PWG5100.11] para 11.5.2
MediaHoleCount	int	number of predrilled holes for the media	para 3.13.6
MediaInfo	string	information that helps describe the media instance	para 3.13.3
MediaKey	string	unique name of the media. The media that MediaKey represents is a named collection of MediaCol elements and their values	para 3.13.1
MediaOrderCount	int	number of sheets, within an ordered sequence of sheets; after which the sequence begins to repeat	para 3.13.7
MediaPreprinted	keyword	pre-printed characteristics of the media *MediaPreprintedWKV*, MediaNsExtensionPattern	Para 3.13.5
MediaRecycled	keyword	recycled characteristics of the media *MediaRecycledWKV*, MediaNsExtensionPattern	para 3.13.1
MediaSize	complex	numerical media width and height dimensions of media in hundredths of a millimeter	para 3.13.8
XDimension	int		para 3.13..8.1
YDimension	int		para 3.13..8.2
MediaSizeName	keyword	Name of medium size that the Printer uses for all impressions of the Job *MediaSizeNameWKV*, MediaSizeNameExtensionPattern	[PWG5101.1] para 5
MediaThickness	int	thickness of the media in .01 millimeters	[PWG5100.11] para 11.5.3
MediaTooth	keyword	The tooth (or roughness) of the media *MediaToothWKV*, MediaNsExtensionPattern	[PWG5100.11] para 11.5.1
MediaType	keyword	t medium type used for all impressions of the Job *MediaTypeWKV*, MediaTypeExtensionPattern	para 3.13.2
MediaWeightMetric	int	weight of the desired media rounded to the nearest whole number of grams per square meter	para 3.13.9
Any	various	Extension point for MediaCol	

5.2.2 Job Ticket Job Description

Job Description as related to a specific Job has a structure like that described for Service JobDescriptionCapabilities para. 4.3.3 except that in a Ticket, the values of the simple Elements correspond to actual values for the given Job rather than describing the capabilities of the Service.

The Job Description complex Element consists of two sequences.

- Elements inherited from the Imaging Job super class (i.e., Imaging Job Description), as represented in Figure 98 and listed in Table 66.
- Service specific extensions to the super class including the extension point for vendors. Those specific to Print Service (and also largely applicable to EmailOut and FaxOut) are represented in Figure 99 and listed in Table 67.

The group of Job Description Elements common to more than one Service that may be included in a Job Ticket are described in Table 67. This table includes both those Elements from the general Imaging Job Description and common Service-specific Elements. Note that, although the Element names are the same or similar to those in Chapter 4, the data types of the Elements are typically different.

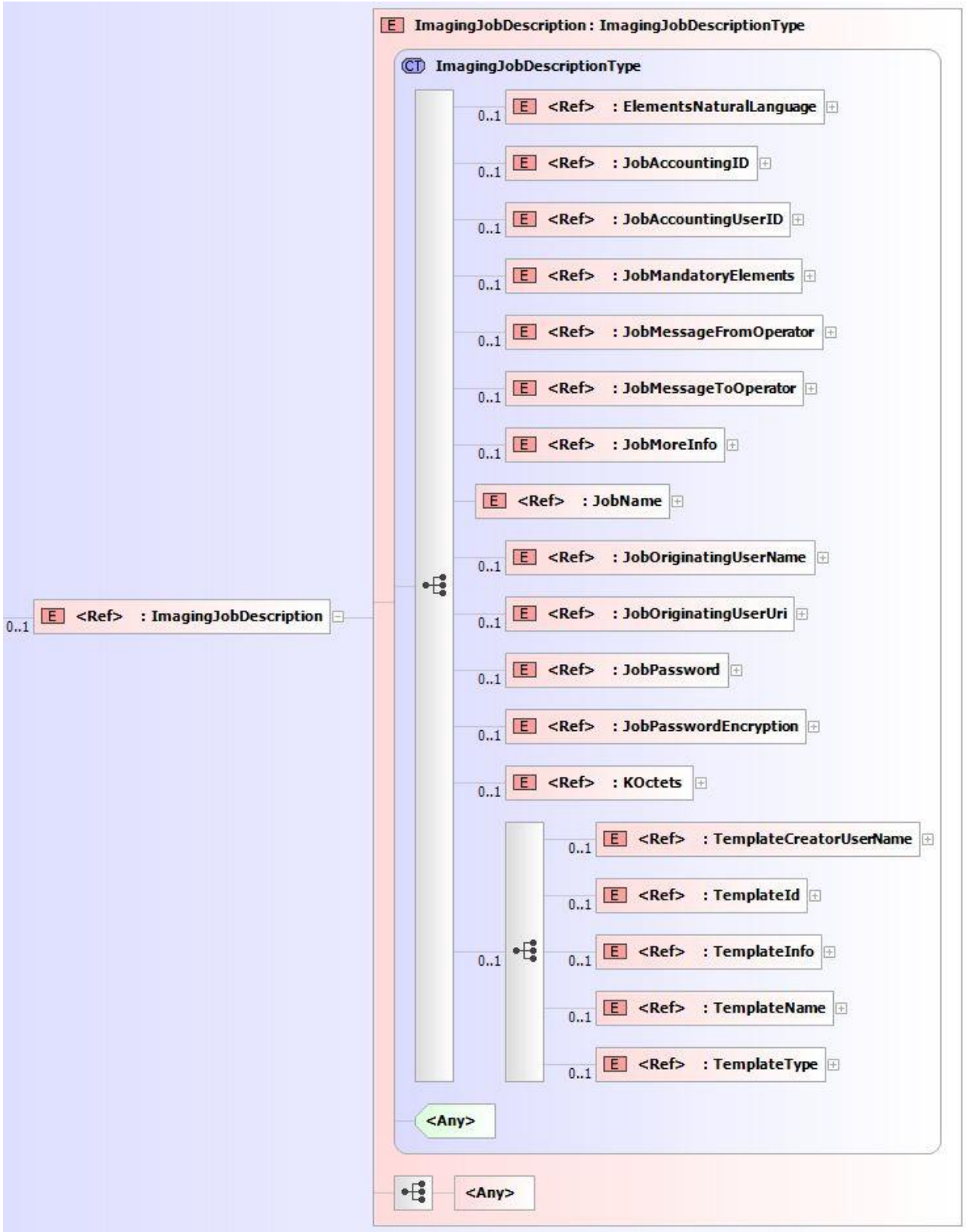

Figure 98 ImagingJobDescription (Superclass)

Table 66 JobTicket ImagingJobDescription Elements (Superclass)

Element	DataType	Description	Reference
ElementsNaturalLanguage	keyword	natural language of the system-generated string Elements, as set by the End User. *NaturalLanguageWKV*	[RFC3066].
JobAccountingID	string	accounting ID associated with this Job.	PWG5100.3] para 3.6
JobAccountingUserID	string	This Element specifies the User ID associated with the "JobAccountId".	[PWG5100.3] para 3.7
JobMandatoryElements	various	List of PWG standard vendor-specified Elements considred mandatory,with each Element of appropriatestatype. Identifies Ticket Elements the Scanner must honor. The Service rejects the request for Job creation if any of the listed Elements are unsupported or contain values that the Service does not support. All of the remaining supplied Elements are best effort.	attribute-fidelity [RFC2911] para 15.1 and [PWG5100.5] para 8.1.1
JobMessageFromOperator	string	message to the End User indicating the reasons for any management action taken on this Job	[RFC2911] para 4.3.16
JobMessageToOperator	string	message from the End User to indicate something about the processing of this Job.	[PWG5100.3] para 3.10
JobMoreInfo	anyUri	User supplied URI referencing some resource with more information about this Job	[RFC2911] para 4.3.4
JobName	string	Service sets this to the client-supplied, end-user friendly name for the Job. When it is not supplied by the client, the Service must generate a name from other information.	[RFC2911] para 4.3.5
JobOriginatingUserName	string	set to the most authenticated printable name of the originating user that the service can obtain	[RFC2911] para4.3.6
JobOriginatingUserUri	anyUri	URI of the User originating the Job	
JobPassword	octetString	password supplied by the client, encrypted according to method specified by the client in the JobPasswordEncryption Element.	[PWG5100.11] para 6.1
JobPasswordEncryption	keyword	encryption the client is using for the supplied value of the JobPassword Element. *JobPasswordEncryptionWKV, KeywordNsExtensionPattern*	PWG5100.11] para 6.2
KOctets	int	total size of this Job's Digital Document(s) in integral units of 1024 octets.	[RFC2911] para 4.3.17.1
TemplateCreatorUserName	string	name of user who created the template used to create the Job Ticket (if any). Note: For a Job Template this is used to hold the most authenticated user name of the Template Creator.	[PWG5108.2] para 7.1.2.2.12
TemplateId	URI	ID of the template used to create the Job Ticket (if any).	[PWG5108.2] para 7.1.2.2.13
TemplateInfo	string	TemplateInfo for the template used to create the Job Ticket (if any).	[PWG5108.2] para 7.1.2.2.14
TemplateName	string	name of template used to create the Job Ticket (if any). Note: For a Job Template this is used to hold the Template Repository wide unique Template name.	[PWG5108.2] para 7.1.2.2.15
TemplateType	keyword	type of the template used to create the Job Ticket (if any). The type of the template MUST be appropriate for the service *TemplateTypeWKV*	[PWG5108.2] para 7.1.2.2.16
Any (service specific)	various	Extension point for JobTicketJobDescription	

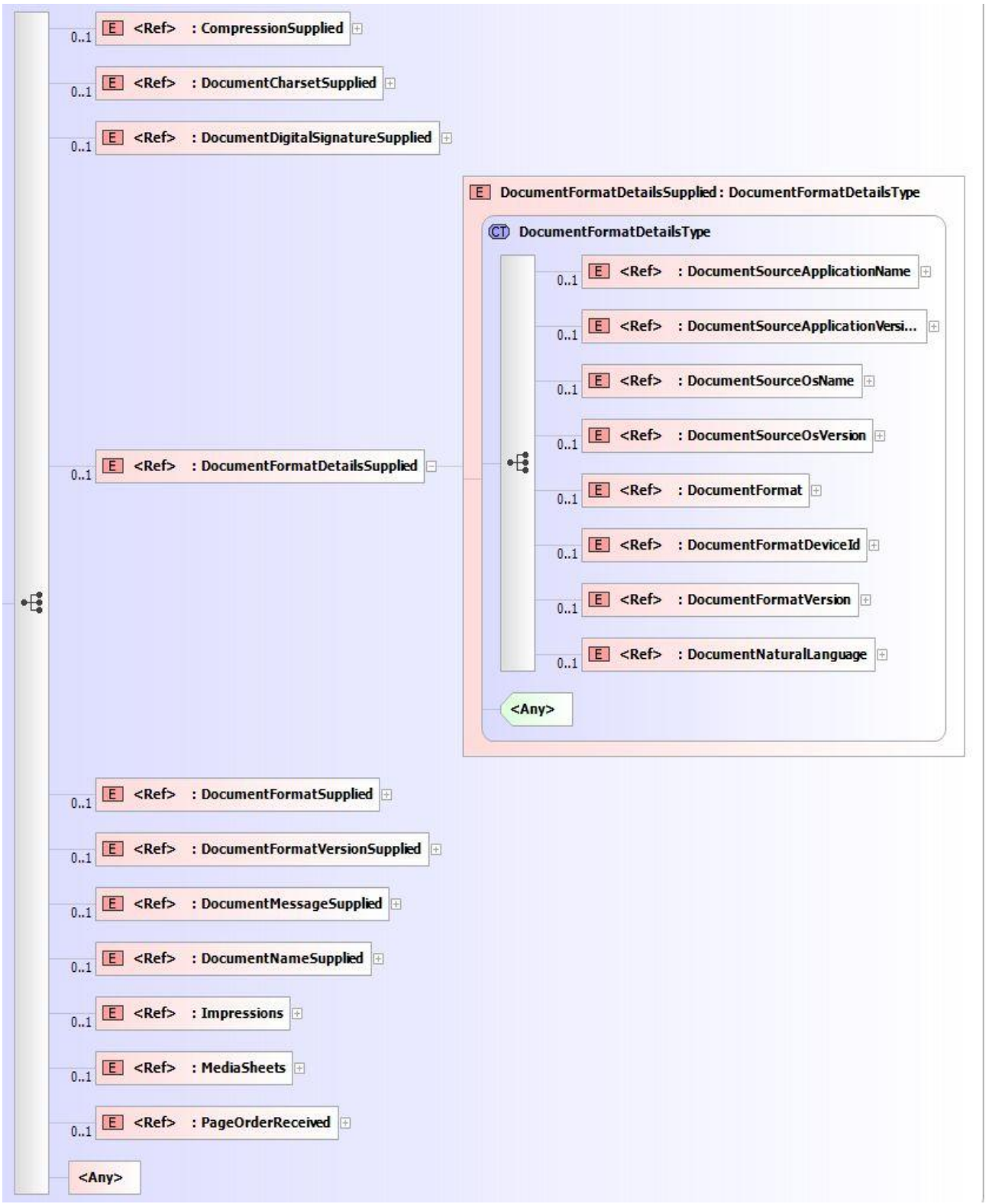

Figure 99 Representative Service-Specific Job Ticket Job Description Elements (Print Service)

Table 67 Service-Specific JobTicketJobDescription Elements

Element	DataType	Description	Reference
CompressiuonSupplied	keyword	Default compression algorithm used for the Documents Data , *CompressionWKV*, KeywordNsExtensionPattern	[PWG5100.7] para5.2.1
DocumentCharsetSupplied	keyword	The default charset of the Documents content , *CharsetWKV*, KeywordNsExtensionPattern	[PWG5100.7] para5.2.2
DocumentDigitalSignatureSupplied	keyword	The type of digital signature used in the Document Content , *DigitalSignatureWKV*, KeywordNsExtensionPattern	[PWG5100.7] para5.2.3
DocumentFormatDetailsSupplied	complex	Summarizes the default distinct contained document formats	[PWG5100.7] para5.2.4
DocumentSourceApplicationName	string	name of the application that created the document, without its version number	[PWG5100.5] para9.1.13
DocumentSourceApplicationVersion	string	version of the application that created the document, without its name	[PWG5100.5] para9.1.13
DocumentSourceOsName	keyword	name of the operating system, without version number, on which the document was generated	[PWG5100.5] para9.1.13
DocumentSourceOsVersion	string	version of the operating system, without its name, on which the document was generated	[PWG5100.5] para9.1.13
DocumentFormat	keyword	Document format (i.e., PDL) for this Document, *DocumentFormatWKV*, MimeExtensionPattern	[RFC2911] para3.2.1.1 [PWG5100.5] para9.1.12
DocumentFormatDeviceId	string	type of device for which the document was formatted, following the IEEE 1284-2000 Device ID string	[PWG5100.5] para9.1.13
DocumentFormatVersion	string	level or version of the DocumentFormat	[PWG5100.5] para9.1.16
DocumentNaturalLanguage	keyword	primary Natural Language of the Document, *NaturalLanguageWKV, NatLangExtensionPattern*	[RFC2911] para3.2.1.1 [PWG5100.5] para9.1.22
DocumentFormatSupplied	keyword	default Document format (i.e., PDL) for Documents in the Job , *DocumentFormatWKV*, KeywordNsExtensionPattern	[PWG5100.7] para5.2.5
DocumentFormatVersionSupplied	string	default level or version of the DocumentFormat	[PWG5100.7] para5.2.6
DocumentMessageSupplied	string	message from either (1) the user to the operator about the Documents or (2) from the operator, system administrator, or "intelligent" process to indicate to the end user the reasons for modification or other management action taken on the Documents	[PWG5100.7] para5.2.7
DocumentNameSupplied	string	default name for the Documents in the Job to be used in an implementation specific manner	[PWG5100.7] para5.2.8
Impressions	int	total size in number of impressions in all the Job's Document(s)	[RFC2911] para4.3.17.2
MediaSheets	int	total number of media sheets to be produced for this Job's Document(s)	[RFC2911] para4.3.17.3
PageOrderReceived	keyword	the order of pages in this Document data as supplied with the job , *PageOrdertWKV*, KeywordNsExtensionPattern	[PWG5100.3] para3.16
Any (service specific)	various	Extension point for JobTicketJobDescription	

5.2.3 JobTicket Job Processing

JobTicketJob Processing provides the Job processing instructions that have been requested by the End User. Each Element has a Must Honor Element. When the value of Must Honor is true, the Service does not process the Job unless the Element is supported; otherwise the Service processes the Job with its best effort.

The Job Processing consists of two sequences:

- Elements inherited from the Imaging Service super class (Figure 100) including Elements such as Job Hold Until. These Elements are described in Table 68.
- Service-specific extensions to the super class. The FaxOut Service and the Copy Service include Job Processing Elements which are also applicable to other Services. These Elements represented in Figure 101, Figure 102, Figure 103 and Figure 104. Service-specific Job Processing Elements are described in Table 69. Note that many of the complex Element structures also appear elsewhere.

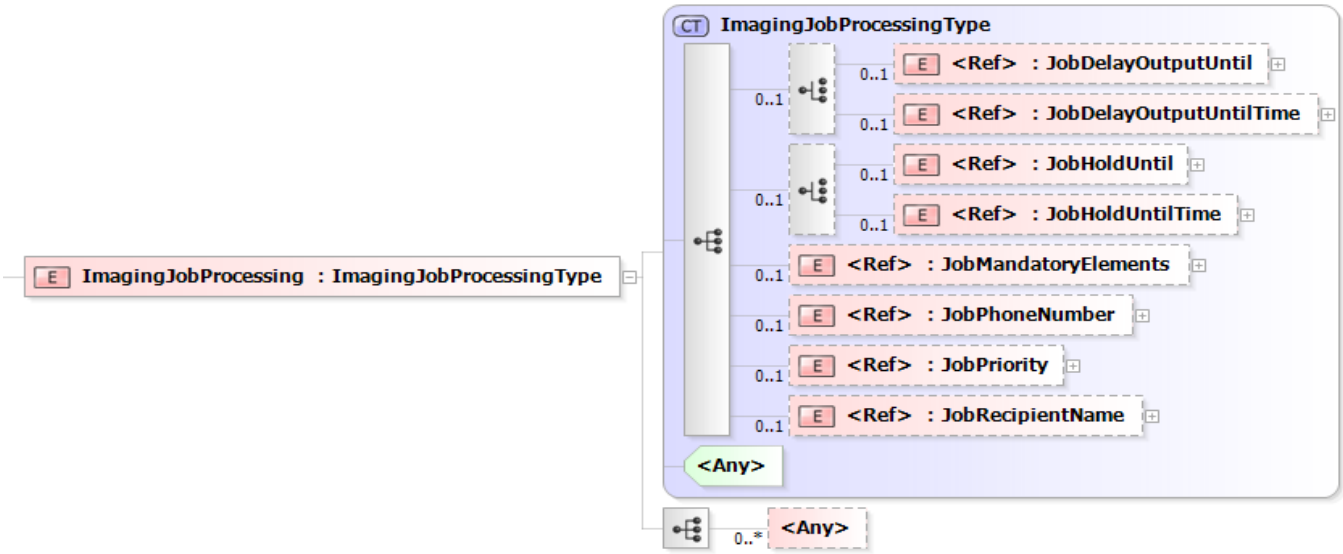

Figure 100 Superclass Job Processing Elements

Table 68 Superclass JobProcessing Element Descriptions

Element	DataType	Description	Reference
JobDelayOutputUntil	keyword	duration of time that a Job output is to be delayed.. *HoldUntilWKV*	[PWG5100.11]para 7.4
JobDelayOutputUntilTime	dateTime	absolute date and time a Job output delay is to expire. This Element allows delay of a Job output until a specific time	[PWG5100.11]para 7.5
JobHoldUntil	keyword	duration of time that a Job is put on hold.. HoldUntilWKV	[RFC2911] para 4.2.2
JobHoldUntilTime	dateTime	absolute date and time a Jos Element allows you to hold a remotely submitted Job until a specific time for processing.	[PWG5100.11] para 5.4
JobPhoneNumber	uri	contact phone number for the Job owner. Can be used to contact the owner in the event additional information is required	[PWG5100.11] para5.5, [RFC3966], [RFC4355]
JobMandatoryElements	various		
JobPriority	int	JobPriorityType	[RFC2911] para4.2.1
JobRecipientName	string	name of the intended recipient of the Job.	[[PWG5100.11para 5.6

5.2.3.1 Description of DestinationUris' Elements

This Job Processing complex element specifies destinations for the Job transmission (e.g., FaxOut). It is a sequence (type:complex) of DestinationUrisEntry each of which defines a single destination. Dial Strings were dropped when RFC3966 superseded RFC2806 [RFC2806], "Dial strings" are the actual numbers, symbols, and pauses entered by a user to place a phone call. We have added some elements to contain the Dial String information even though they could be encoded as RFC3966 compliant parameters.

5.2.3.1.1 DestinationUri

. The DestinationUri is an RFC3966 [RFC3966] compliant URI. The abstract datatype is string. The corresponding Job Ticket Capabilities element for this element is DestinationUriSupported and is a set of strings. Each string contains the URI scheme supported fo use in DestinationUri.

5.2.3.1.2 PreDialString

The PreDialString is the Dial string entered before the DestinationUri is applied. The data type for a dial string is a string. The ABNF that applies to a PreDialString is given below. The corresponding Job Ticket Capabilities element is of type Boolean that indicates whether or not this element is supported.

```
DialString      = 1*(phonedigit / dtmf-digit / pause-character)
pause-character    = one-second-pause / wait-for-dial-tone
one-second-pause   = "p"
wait-for-dial-tone  = "w"
dtmf-digit      = "*" / "A" | "B" / "C" / "D" / "#"
phonedigit      = DIGIT / [ visual-separator ]
visual-separator   = "-" / "." / "(" / ")" / "" / "(" / ")"
```

5.2.3.1.3 PostDialString

The PostDialString is the Dial string entered after the DestinationUri is applied. The data type for a dial string is a string. The Dial Sting ABNF (abiove) also applies to a PostDialString. The corresponding Job Ticket Capabilities element is of type Boolean that indicates whether or not this element is supported.

5.2.3.1.4 T33Subaddress

The T33 subaddressing can be specified by the T33Subaddress element. Because a T33 subaddress can only contain digits [RFC3192], the datatype for the T33Subaddress is an integer. Multiple T33 subaddresses are not supported in a single DestinationUrisEntry. If multiple mailboxes are to be addressed, each one requires its own entry. The corresponding Job Ticket Capabilities element is of type Boolean that indicates whether or not this element is supported.

5.2.3.1.5 Any

This is the extension point for the DestinationUris element.

5.2.3.2 Description of ConfirmationSheetPrint Element

This optional Job Processing element specifies whether or not a confirmation sheet will be printed upon job completion. Services such as FaxOut offer such a feature. The data type for this element is Boolean. If ConfirmationSheetPrint is true, a confirmation sheet will be printed when the job reaches a terminating state. The format of the confirmation sheet is implementation specific. The content of the confirmation is outside the scope of this specification but is expected to conform to common practive and/or prevailing

regulatory requirements. The corresponding Job Ticket Capabilities element is of type Boolean that indicates whether or not this element is supported.

5.2.3.3 Description of CoverSheetInfo's Elements

This optional Job Processing element contains user supplied information to be put on a system generated Fax cover sheet. Note that the syntax for all the contained elements in the corresponding Job Ticket Capabilities elements are Boolean and represent the implementations support for the Job Processing elements.

5.2.3.3.1 CompanyName

This Job Processing element contains the user supplied string to place in the "Company Name" field of the system generated Fax Cover Sheet.

5.2.3.3.2 DateTime

This Job Processing element contains the date and time to place in the "Date" and "Time" fields of the system generated Fax Cover Sheet. Note this does not affect the date/time stamp contained in the Fax Protocol itself.

5.2.3.3.3 From

This Job Processing element contains a user supplied string to be placed in the "From" field of the system generated Fax Cover Sheet.

5.2.3.3.4 Logo

This Job Processing element contains a URL of an image to be placed in an appropriate location of the system generated Fax Cover Sheet.

5.2.3.3.5 Message

This Job Processing element contains the user supplied string to place in the "Message" field of the system generated Fax Cover Sheet.

5.2.3.3.6 Subject

This Job Processing element contains the user supplied string to place in the "Subject" field of the system generated Fax Cover Sheet.

5.2.3.3.7 To

This Job Processing element contains the user supplied string to place in the "To" field of the system generated Fax Cover Sheet.

5.2.3.3.8 Any

This is the extension point for the CoverSheetInfo element.

5.2.3.4 Description of RetryInfo's Elements

This Job Processing element contains user supplied information to be used to control the attempts to complete a job after an initial failure to transmit a document.

5.2.3.4.1 NumberOfRetries

This Job Processing element specifies the number of attempts to retransmit a document in the event of transmission failure, before aborting the job. The data type for this element is an integer. The corresponding Job Ticket Capabilities element is a range of integers specifying the allowed values for number of attempts.

5.2.3.4.2 RetryInterval

This Job Processing element specifies the time duration in seconds between transmission attempts. The data type for this element is an integer. The corresponding Job Ticket Capabilities element is a range of integers specifying the allowed values for time duration in seconds between transmission attempts.

5.2.3.4.3 RetryTimeout

This Job Processing element specifies the time duration in seconds before terminating a transmission attempt when no document data is being sent. . The data type for this element is an integer. The corresponding Job Ticket Capabilities element is a range of integers specifying the allowed values for time duration in seconds before terminating a transmission attempt when no document data is being sent.

5.2.3.4.4 Any

This is the extension point for the RetryInfo element.

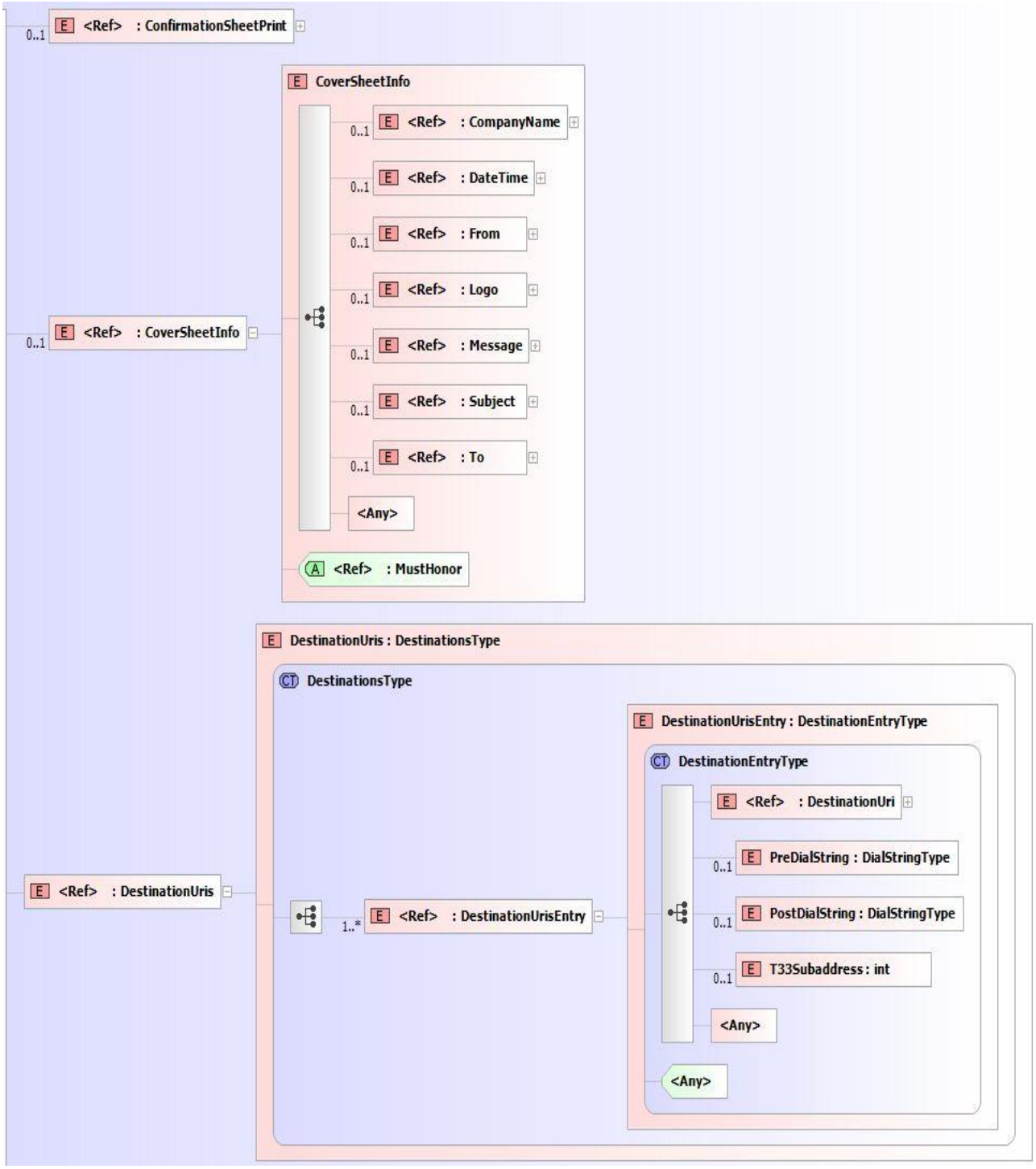

Figure 101 Representative Service-Specific Job Processing Elements (FaxOut Service) Sheet 1

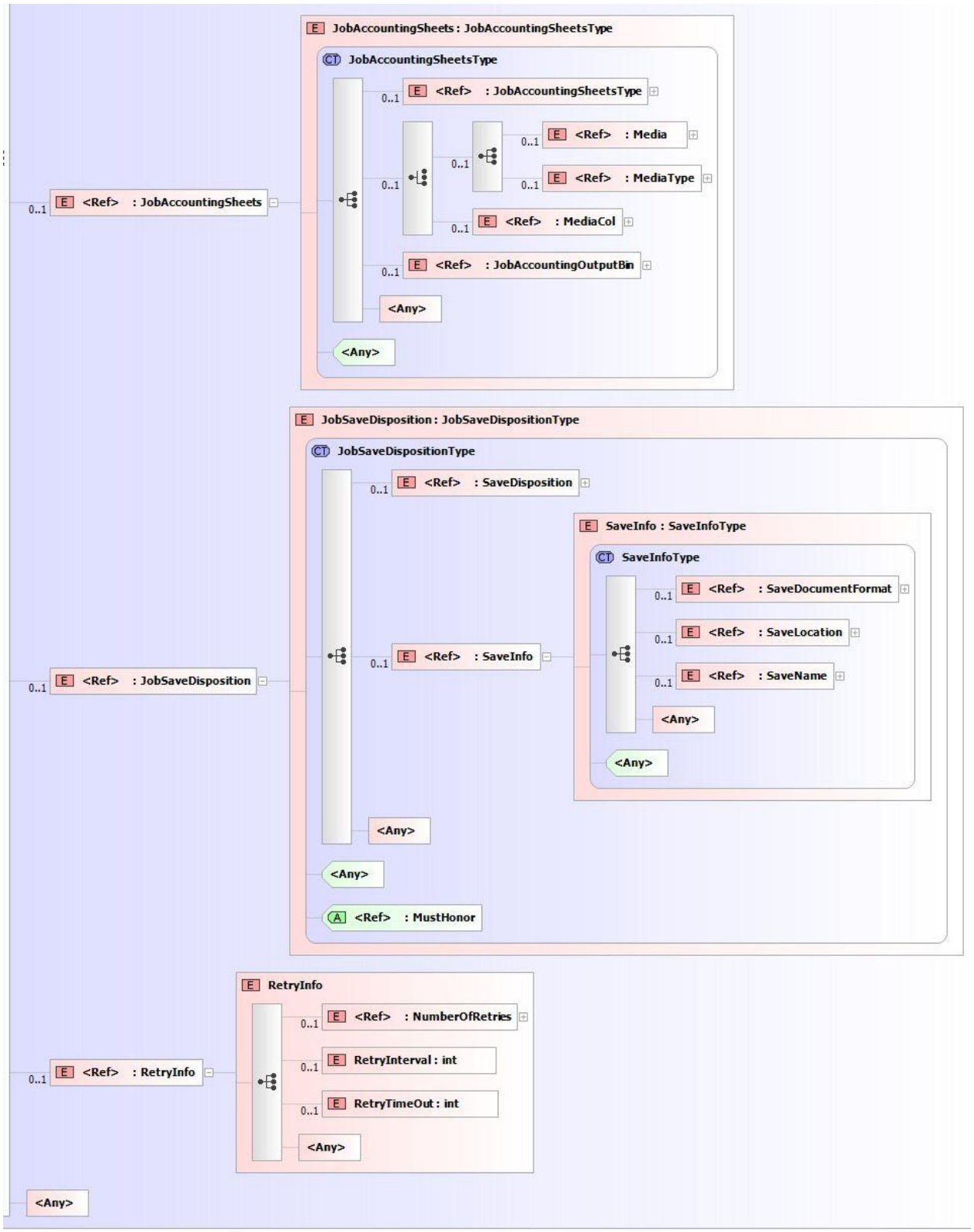

Figure 102 Representative Service-Specific Job Processing Elements (FaxOut Service) Sheet 2

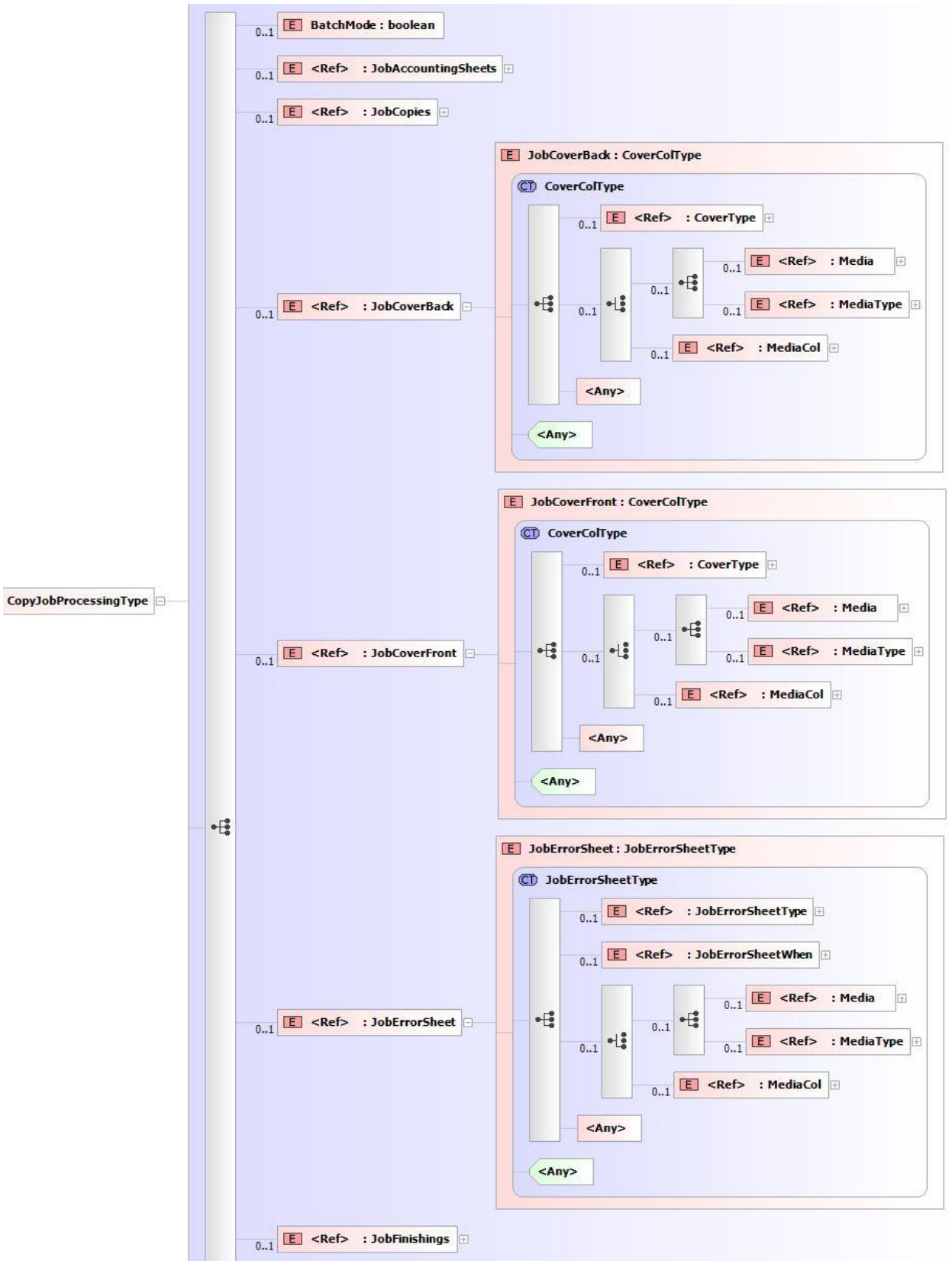

Figure 103 Representative Service-Specific Job Processing Elements (Copy Service) Sheet 1

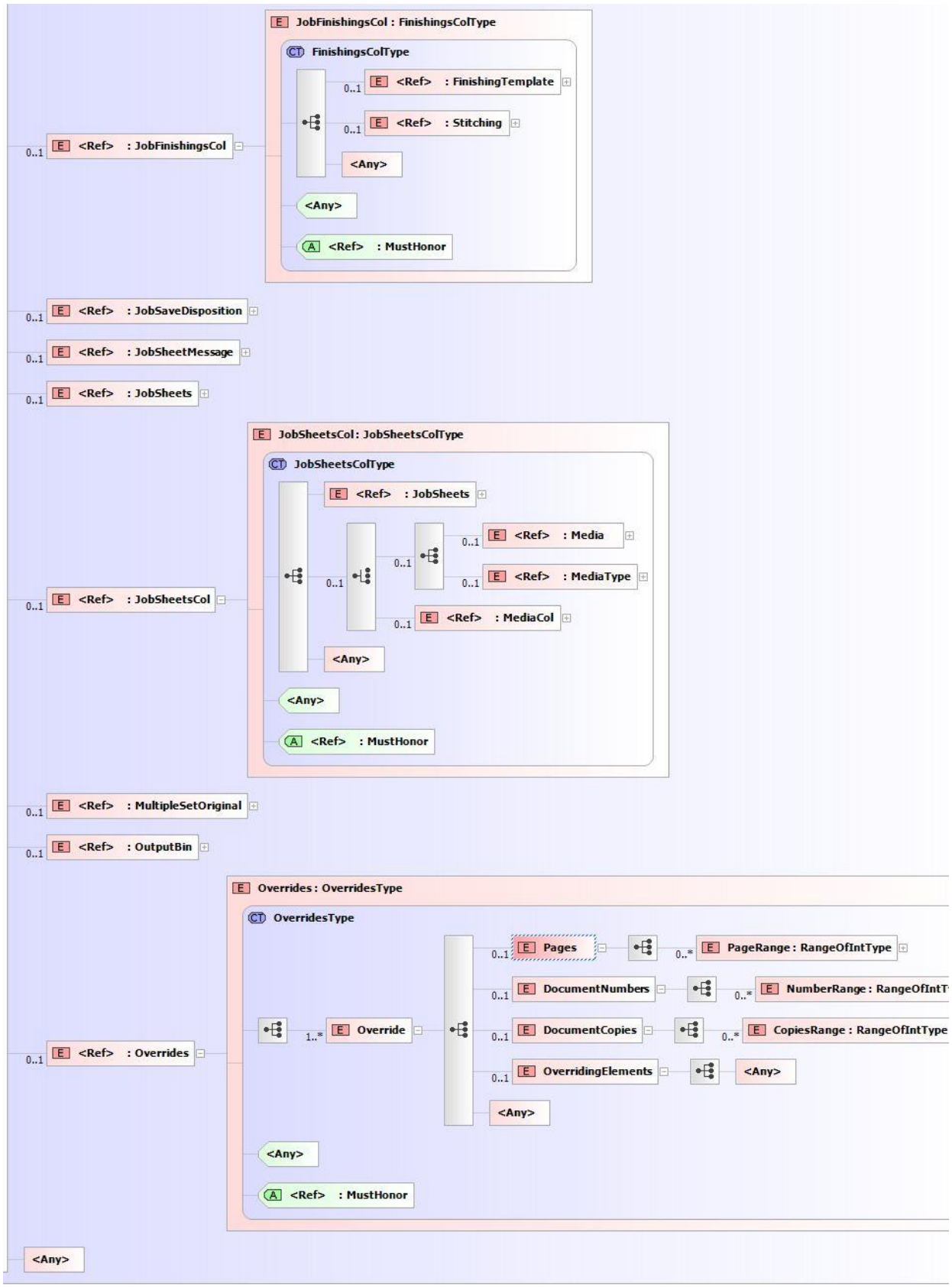

Figure 104 Representative Service-Specific Job Processing Elements (Copy Service) Sheet 2

Table 69 Service-Specific Job Processing Elements

Element	DataType	Description or *Keyword*	Reference
BatchMode	boolean	job is to be processed in Batch Mode. Batch mode for each Service, if applicable, is defined in the specification of the Service.	[PWG5108.2] para 8.1.3.3.5
ConfirmationSheetPrint	boolean	if true, confirmation sheet, for example for a FaxOut Job, will be printed when the job reaches a terminating state. The format of the confirmation sheet is implementation specific.	See 5.2.3.2
CoverSheetInfo	complex	Information entered by user to be put on Fax cover sheet.	See 5.2.3.3
CompanyName	string	user-supplied company name	See 5.2.3.3.1
DateTime	string	user-supplied date and time	See 5.2.3.3.2
From	string	user supplied name of the person group or entity for the cover sheet	See 5.2.3.3.3
Logo	URI	user supplied URL pointing to an image to use as the logo for the cover sheet	See 5.2.3.3.4
Message	string	user-supplied message for the cover sheet	See 5.2.3.3.5
Subject	boolean	user-supplied subject entry for the cover sheet	See 5.2.3.3.6
To	boolean	user-supplied name of the intended recipient of the document	See 5.2.3.3.7
Any	various	extension point for CoverSheetInfo	See 5.2.3.3.8
DestinationUris	Complex	see para.5.2.3.1	
DestinationUri	URI	URL used to transfer a Digital Document to its Destination.	See 5.2.3.1
PreDialString	string	dial string entered before the DestinationUri is applied	See 5.2.3.1
PostDialString	string	dial string entered after the DestinationUri is applied	See 5.2.3.1.3
T33Subaddress	int	For T33 subaddressing	See 5.2.3.1.4
Any	various	extension point for DestinationUris	
DocumentOutputMode	keyword	Way in which the output documents are partitioned among files. See para 2.4.2 *DocumentOutputModeWKV,* KeywordNsExtensionPattern	[PWG5108.2] para 8.1.3.3.6
JobAccountingSheets	complex	Specifies the accounting sheet for a job	[PWG5100.3] para 3.8
JobAccountingSheetsType	keyword	Specifies the accounting sheet format *JobAccountingSheetTypeWKV,* KeywordNsExtensionPattern	[PWG5100.3] para 3.8.1
Media	keyword	Name of media to be used for accounting sheet, *MediaWKV*, KeywordNsExtensionPattern	[RFC2911] para 4.2.11
MediaType	keyword	medium type used for the job accounting sheet *MediaTypeWKV,* KeywordNsExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	complex	See Media Collection (Specific) 5.2.1.1	[PWG5100.3] para 3.13
JobAccountingOutputBin	keyword	output bin where the accounting sheet is to be placed, *OutputBinWKV,* KeywordNsExtensionPattern	[PWG5100.3] para 3.8.3
Any	various	extension point for JobAccountingSheets	
JobCopies	int	The number of copies of the Job to be printed	[PWG5100.7] para 4.1.1
JobCoverBack	complex	Characteristics of back cover for this Job	
CoverType	keyword	Indicates if back cover is requested and which sides will contain print stream pages, *CoverTypeWKV*, KeywordNsExtensionPattern	[PWG5100.3]para3.1.2
Media	keyword	Name of mediato be used for back cover *MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	keyword	medium type to be used for back cover *MediaTypeWKV,* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	complex	See 5.2.1.1	[PWG5100.3] para 3.13

Element	DataType	Description or *Keyword*	Reference
Any	various	Extension point for JobCoverBack	
JobCoverFront	complex	Characteristics of front cover for this Job	[PWG5100.7]para4.1.3
Coverype	keyword	Indicates if front cover is requested and which sides will contain print stream pages, *CoverTypeWKV*, KeywordNsExtensionPattern	[PWG5100.3]para3.1.2
Media	keyword	Name of mediato be used for front cover *MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	keyword	medium type to be used for front cover *MediaTypeWKV,* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	complex	See 5.2.1.1	[PWG5100.3] para 3.13
Any	various	Extension point for JobCover Front	
JobErrorSheet	complex	Characteristics of JobErrorSheet for this Job	[PWG5100.3] para 3.9
JobErrorSheetType	keyword	Indicates if JobErrorSheet is requested and which sides will contain print	[PWG5100.3] para 3.9.1
JobErrorSheetWhen	keyword	Indicatedsconditions to produce JobErrorSheet *JobErrorSheetTypeWKV,* KeywordNsExtensionPattern	
Media	keyword	Name of mediato be used for JobErrorSheet *MediaSizeAliasNameWKV, MediaSizeLegacyNamesWKV, MediaSizeSelfDescribingNameWKV* MediaSizeNameExtensionPattern	[RFC2911] para 4.2.11
MediaType	keyword	medium type to be used for front cover *MediaTypeWKV,* MediaTypeExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	complex	See 5.2.1.1	[PWG5100.3] para 3.13
Any	various	extension point for JobErrorSheet	
JobFinishings	keyword	finishing that the Printer uses for each job copy of the Job, *FinishingsWKV*, KeywordNsExtensionPattern	[RFC2911] para 4.2.6 [PWG5100.1] para 2 [PWG5100.7] para 4.1.4
JobFinishingsCol	complex	specify detailed finishing options not possible with the "JobFinishings"	[PWG5100.3] para 3.2
FinishingTemplate	string	string specifying some particular finishing operation	[PWG5100.3] para 3.2.1
Stiching	complex	detailed stitching parameters	[PWG5100.3] para 3.2.2
StitchingReferenceEdge	keyword	stitching reference edge of the output media *StitchingReferenceEdgeWKV, KeywordNsExtensionPattern*	[PWG5100.3] para 3.2.2.1
StitchingOffset	int	perpendicular distance from the reference edge to the stitching axis in hundredths of a millimeter.	[PWG5100.3] para 3.2.2.2
StitchingLocations	int	distance along the stitching axis where a stitch will be placed in hundredths of a millimeter	[PWG5100.3] para 3.2.2.3
Any	various	extension point for Stiching	
Any	various	extension point for JobFinishingsCol	
JobSaveDisposition	complex	used to archive/save the Document Data of a job	[PWG5100.11] para 7.9
SaveDisposition	keyword	specifies whether or not the job MUST be printed and/or saved, *SaveDispositionWKV,* KeywordNsExtensionPattern	[PWG5100.11] para 7.9.1
SaveInfo	complex		[PWG5100.11] para 7.9.1.2
SaveDocumentFormat	keyword	SaveDocumentFormatType *DocumentFormatWKV, MimeExtensionPattern*	[PWG5100.11] para 7.9.1.2.3
SaveLocation	uri	path to the directory as a URI where the Service saves the Document Data and other information	[PWG5100.11] para 7.9.1.2.1
SaveName	string	name of the saved job	[PWG5100.11] para 7.9.1.2.2
Any	various	Extension point for SaveInfo	

Element	DataType	Description or *Keyword*	Reference
Any	various	Extension point for JobSaveDisposition	
JobSheetMessage	string	a message that is delivered with the job	[PWG5100.3] para 3.12
JobSheets		Specifies which Job start/end sheet(s), will be printed with a Job *JobSheetsWKV, KeywordNsExtensionPattern*	[RFC2911] para 4.2.3 [PWG5100.3] para 6.2
JobSheetsCol	complex	specify the media for the JobSheet	[PWG5100.3] para 3.11
JobSheets		Specifies which job start/end sheet(s), will be printed with a Job *JobSheetsWKV, KeywordNsExtensionPattern*	[RFC2911] para 4.2.3 [PWG5100.3] para 6.2
Media	keyword	Name of media to be sued for JobSheets *MediaWKV*, KeywordNsExtensionPattern	[RFC2911] para 4.2.11
MediaType	keyword	medium type used for JobSheets *MediaTypeWKV,* KeywordNsExtensionPattern	[PWG5100.3] para 3.13.2
MediaCol	complex	See Media Collection (Specific) 5.2.1	[PWG5100.3] para 3.13
Any	various	Extension point for JobSheetsCol	
MultipleDocumentsHandling	keyword	*MultipleDocumentsHandlingWKV,* KeywordNsExtensionPattern	[RFC2911] para 4.2.4
MutipleSetOriginal	boolean	Service is required to intake multiple sets of originals from the selected inpb will be released after being put on hold. Thiut source.	[PWG5108.2] para 8.1.3.3.7
OutputBin	keyword	output bin where the Job is to be placed *OutputBinWKV*, KeywordNsExtensionPattern	[PWG5100.2] para 2.1 [PWG5100.5] para 8.1
OutputDevice	string	Requested OutputDevice	[PWG5100.7] para 4.2.1 [PWG5100.5] para 8.1
Overrides	complex	Page processing overrides	[PWG5100.6] para 4.1
Pages	range of int	identifies one or more pages where the override will be applied	[PWG5100.6] para 4.1.1
DocumentNumbers	range of int	identifies one or more documents in which the pages will have the override applied	[PWG5100.6] para 4.1.2
DocumentCopies	range of int	identifies one or more copies of the document in which the pages will have the override applied	[PWG5100.6] para 4.1.3
OverridingElements	list of document ticket elements	Document Ticket processing elements to apply to the specified Document pages	[PWG5100.6] para 4.1.4
Any	various	Extension point for Overrides	
PagesPerSubset	int	Combines all of the Pages of all of the Documents into a single stream of -Pages. Then the Printer partitions that single stream into contiguous subsets of -Pages according to the value of this Element	[PWG5100.4] para 5.3
RetryInfo	complex	Supported capabilities of elements used to control document transmission timeouts	See 5.2.3.4
NumberOfRetries	int	number of retries before Job aborts	See 5.2.3.4.1
RetryInterval	int	time duration in seconds between retry attempts	See 5.2.3.4.2
RetryTimeout	int	time duration in seconds before terminating a retry	See 5.2.3.4.3
Any (service specific)	various	an extension point for RetryInfo	See 5.2.3.4.4
Any	various	Extension point for Job Processing	

5.3 Job Receipt

This Element has exactly the same structure as the Job Ticket. For each processing Element of a Job, it records the actual value used by the Service for processing the Job. It contains the Elements supplied by the Client and applied to the Job, any Element or values substitutions made by the Service and any default Elements or values applied by the Service. See paragraph 5.2 for Element descriptions.

6 Document Model

The general Imaging Document model includes the Document Status, Document Ticket and Document Receipt Element groups, as shown in Figure 105. The individual Services that include accessible Document objects each have their own <service>Document model, paralleling this general Imaging Document model. Jobs can contain zero or more Documents. During Job creation, it is possible that temporarily there are zero Documents. Note that some Services do not include accessible <service>Documents.

The state of the Document is described in the Document Status Element. This contains descriptive information about the Document processing instructions. The Document Ticket and the Document Receipt have the same structure, but the Element values in the Document Ticket are the information provided by the user in the CreateJob operation while the Element values in the Document Receipt reflect what was actually used in processing the Document.

Document Tickets allow a User to supply Document Processing Elements to override the Document Processing instructions in the Job Ticket on a Document by Document basis. This Document model mirrors the Job model, described in Chapter 5.

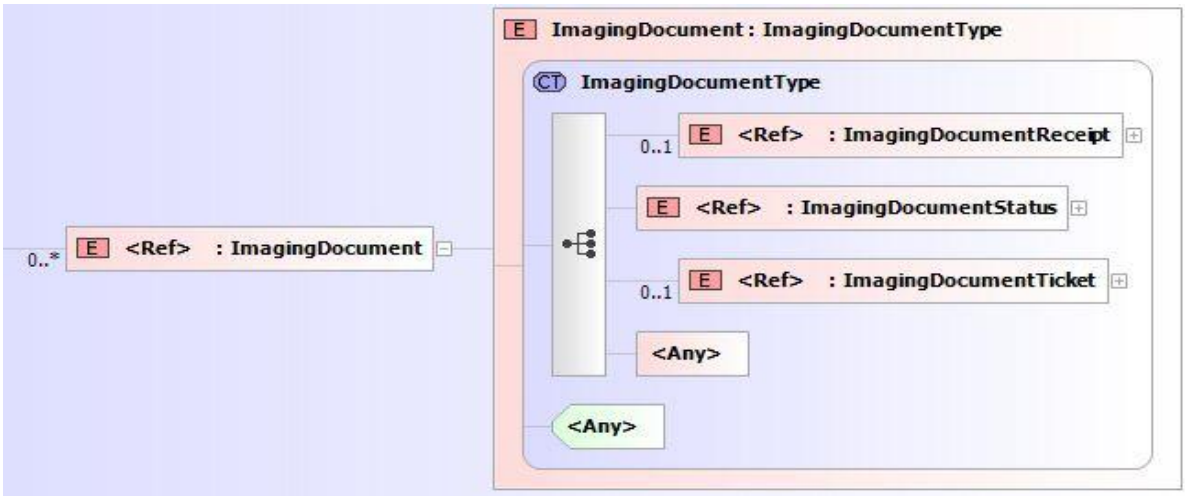

Figure 105 Imaging Document Model

6.1 Document Status

Job Status consists of two sequences.

- Elements inherited from the Imaging Document super class (i.e., Imaging Document Status) See Figure 106 for the structure and Table 70 for the Elements description.
- Service specific extensions to the super class such as the Service counters. Representative Service-specific Elements are represented in Figure 107 and described in Table 71.

Document Status Elements including both those from the abstract Imaging Status and representative Service-specific elements are described in The complex Elements in the table also occur elsewhere. Please refer to the reference sections for the details of these elements.

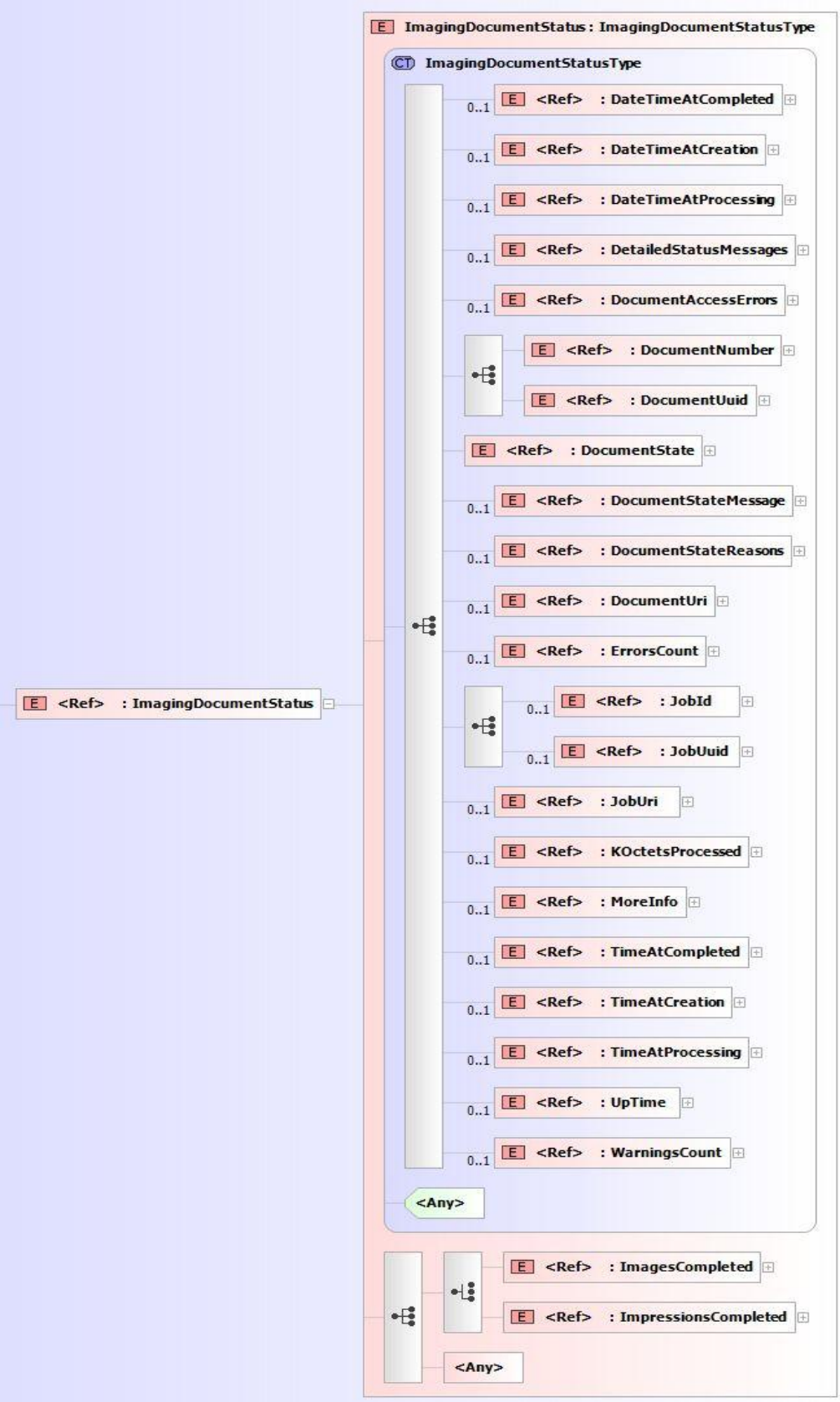

Figure 106 Imaging Document Status

Table 70 Document Status Elements

Element	DataType	Description or *Keyword Group*	Reference
DateTimeAtCompleted	DateTime	date and time at which the Job object completed or was canceled or aborted	[RFC2911] para 4.3.14.7
DateTimeAtCreation	DateTime	date and time at which the Job object created.	[RFC2911] para 4.3.14.5
DateTimeAtProcessing	DateTime	date and time at which the Job object first began processing	[RFC2911] para 4.3.14.6
DetailedStatusMessages	list of string	additional detailed and technical information about the Job. Element	[RFC2911] para 4.3.10
DocumentAccessErrors	list of string	additional information about each Document access error for this Job encountered by the Service after attempting to store or access Document Data at the locations supplied	[RFC2911] para 4.3.11
DocumentNumber	int	uniquely identifies a Document within a Job.	[PWG5100.4] para 9.2, [PWG5100.5] para 9.1.23
DocumentUuid	anyUri	An urn::uuid unique URI value	[RFC4122]
DocumentState	keyword	current state of Document. *DocumentStateWKV*	[PWG5100.5] para 9.1.25
DocumentStateMessage	string	information about the Document State and StateReasons in human readable text.Element.	[PWG5100.5] para 9.1.26
DocumentStateReasons	list of keywords	additional detail about the Document state. The specific keywords allowed are defined within the specification for the Service. *DocumentStateReasonsWKV, KeywordNsExtensionPattern*	[PWG5100.5] para 9.1.27 and {RFC2911] para 4.3.8 for standard values.
DocumentUri	anyUri	An urn::uuid unique URI value	[RFC4122]
ErrorsCount	int	number of errors encountered while processing the Document	[PWG5100.5] para 9.1.29
JobId	int	JobID of the Job to which this Document belongs.	[PWG5100.5] para 9.1.18
JobUuid	anyUri	An urn::uuid unique URI value [RFC4122]	
JobUri	URI	globally unique ID of the Job to which this Document belongs	[PWG5100.5] para 9.1.19
KOctetsProcessed	int	total number of octets processed to that point, in integral units of 1024 octets	[RFC2911] para 4.3.17.1
MoreInfo	string	URI used to obtain information intended for End User consumption about this specific Job	[RFC2911] para 4.3.4
TimeAtCompleted	int	time at which the Document completed, in “<service>UpTime” seconds	[RFC2911] para 4.3.14.3
TimeAtCreation	int	time at which the Document was created in “<service>UpTime” seconds	[RFC2911] para 4.3.14.1
TimeAtProcessing	int	time at which the Document first began processing in “<service>UpTime” seconds	[RFC2911] para 4.3.14.2
UpTime	int		
WarningsCount	int	total number of warnings that a Scan Service has generated while processing and storing the Job’s Documents).	[PWG5100.4 para 6.1
ImagesCompleted	int	Progress measure for Job in terms of output. Output may be Images for Hardcopy Document producers and/or Impressions for Hardcopy Document consumers. See PWG Counter Spec [PWG5106.1]	
ImpressionsCompleted	int		
Any	various	Extension point for DocumentStatus	

Figure 107 Service-Specific Document Status Elements

Table 71 Service-Specific Document Status Elements

Element	DataType	Description or *Keyword Group*	Reference
CurrentPageOrder	keyword	the page order of the pages in the document data, *PageOrderWKV*	[PWG5100.3] para 4.1
DocumentFormatDetailsDetected	complex	See breakdown of constituent elements in Table 67.	
DocumentFormatDetected	keyword	The service sets this to the actual DocumentFormat detected when auto-sensing, DocumentFormatWKV, MimeExtensionPattern	[PWG5100.5] para 9.1.15
DocumentFormatVersionDetected	string	The service sets this to the actual DocumentFormat level detected when auto-sensing	[PWG5100.5] para 9.1.17
Impressions (or Images) Completed	int	Number of Impressions (for Impression output Services) or Images (for Image output Services) processed by the service	[PWG5106.1]
ImpressionsCompletedCurrentCopy	int	number of impressions completed for the current iteration of this Document so far	
MediaSheetsCompleted	int	media-sheets completed marking and stacking for this Document so far	[RFC3381] para 4.4
OutputDeviceAssigned	string	the output device to which the service has assigned this Job	[RFC2911] para 4.3.18.3
SheetsCompletedCopyNumber	int	Number of the copy being stacked for this Document	[RFC2911] para 4.3.13
<service>Counters	complex	counters for the amount of work performed for this Document by the Service, including timers covering utilization, and monitoring information covering errors, warnings traffic, Job counts	[RFC3381] para 4.2
ServiceUri	anyUri	the URI of the Service containing the Job which contains this Document	[PWG5106.1]
Any	various	extension point for<service>DocumentStatus	PWG5100.5 para 9.1.24

6.2 Document Ticket

The Document Ticket contains description and processing Elements provided by the Client. The general Imaging Document Ticket is shown in Figure 108. All <service> Document Tickets follow this form, although the description and processing elements may contain Service specific elements. Document Ticket information is used by the Service during the processing of a Document. The information is made available to Clients through the Get Document Elements operation.

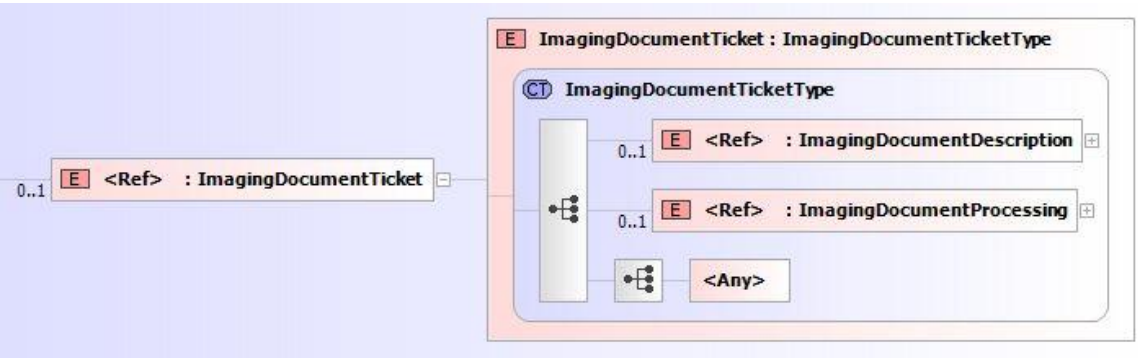

Figure 108 Imaging Document Ticket

6.2.1 Document Description

Document Description consists of two sequences:

- Elements inherited from the Imaging Document abstract class, represented in Figure 109 with the Elements described in Table 72.
- Service specific extensions. Representative Service-specific elements are shown in Figure 110 and described in Table 73. These Elements also appear in several other places in the Model and the structures of the complex Elements is described elsewhere, as referenced in the table.

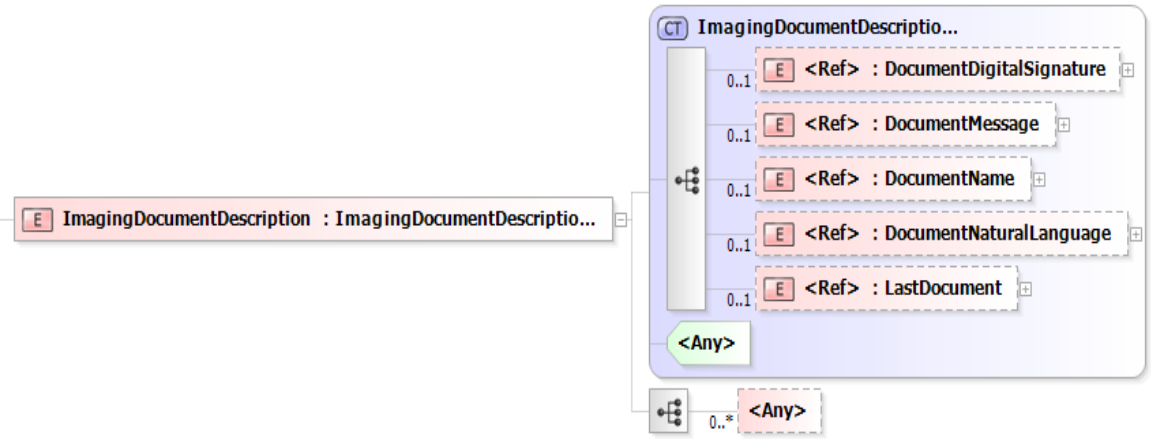

Figure 109 Imaging Document Description

Table 72 Document Description Elements

Element	DataType	Description or *Keyword*	Reference
DocumentDigitalSignature	keyword	type of digital signature that is used in the creation of the Digital Document. *DocumentDigitalSignatureWKV, KeywordNsExtensionPattern*	[PWG5100.7] para 3.2.3 [PWG5100.5] para 9.1.11
DocumentMessage	string	message from either 1) the user to the operator about the Document or 2) from the operator, system administrator, or "intelligent" process to indicate to the End User the reasons for modification or other management action taken on the Document.	[PWG5100.5] para 9.1.20
DocumentName	string	name for this Document to be used in an implementation specific manner.	[RFC2911] para 3.2.1.1
DocumentNaturalLanguage	keyword	hint about the language used in Hard Copy Documents that contain text. This Element is not useful for Hard Copy Documents that do not contain text or are not intended to undergo Optical Character Recognition *NaturalLanguageWKV*	[RFC2911] para3.2.1.1, [PWG5100.5] para 9.1.22
LastDocument	Boolean	last Document in the Job. (Element set to false or omitted for Document which is not the last)	[RFC2911] para 3.3.1
Any	various	extension point for DocumentDescription	

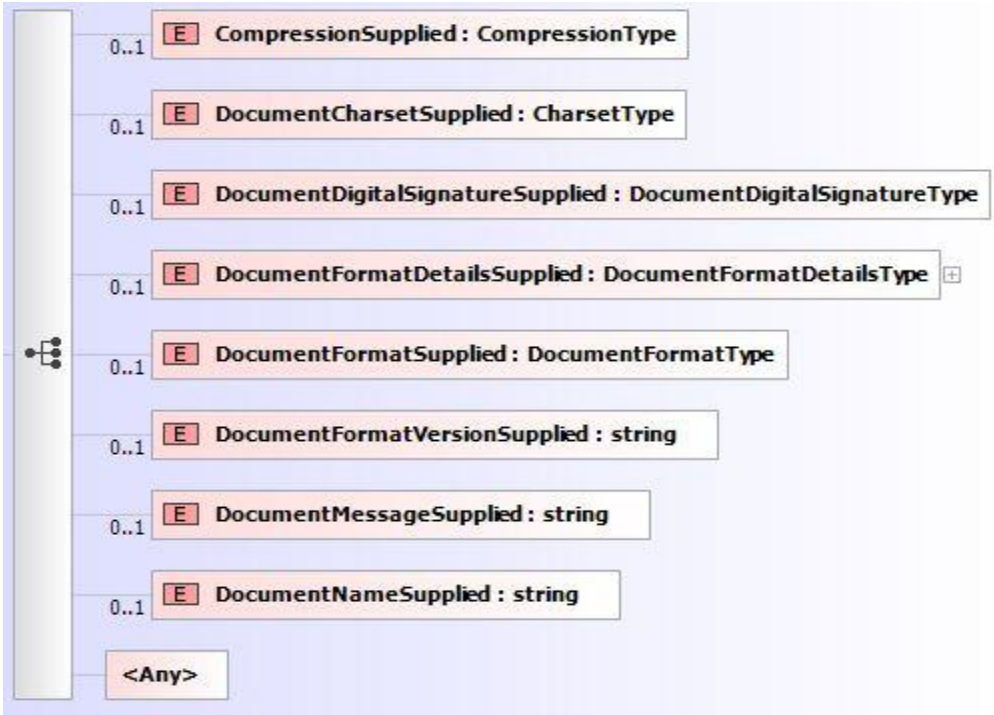

Figure 110 Service-Specific Document Description Elements

Table 73 Service-Specific Document Description Elements

Element	DataType	Description or *Keyword*	Reference [PWG5100.7]
CompressionSupplied	keyword	compression algorithm used for the Documents Data *CompressionWKV*, KeywordNsExtensionPattern	para 5.2.1
DocumentCharsetSupplied	keyword	default charset of the Documents content, CharsetWKV, StringExtensionPattern	para 5.2.2
DocumentDigitalSignatureSupplied	keyword	type of digital signature, if any, used in the Document Content *DocumentDigitalSignatureWKV*, KeywordNsExtensionPattern	para 5.2.3
DocumentFormatDetailsSupplied	complex	Summarizes the distinct contained document format, See Figure 99 and Table 67	para 5.2.4
DocumentFormatSupplied	keyword	The Document format (i.e., PDL) for Document, DocumentFormat, MimeExtensionPattern	para 5.2.5
DocumentFormatVersionSupplied	string	level or version of the DocumentFormat	para 5.2.6
DocumentMessageSupplied	string	message from either (1) the user to the operator about the Documents or (2) from the operator, system administrator, or "intelligent" process to indicate to the end user the reasons for modification or other management action taken on the Document	para 5.2.7
DocumentNameSupplied	string	name for the Document to be used in an implementation specific manner	para 5.2.8
Any	various	Extension point for <service>DocumentDescription	

6.2.2 Document Processing

This group Element has exactly the same structure as the Document Processing Element of Job (See 5.2.1). It provides the Document processing instructions that have been requested by the End User at each Document level, overriding the Job level Document processing instructions.

6.3 Document Receipt

This Service-specific Element has exactly the same structure as the Document Ticket. For each processing Element of a Document, it records the actual value used by the Service for processing the Document. It contains the Elements supplied by the Client, any substitutions made by the Service and any Default Elements applied by the Service. See Section 6.2 for Element descriptions.

7 Service Operations and States

7.1 General Service Sequence of Operation

A Service operates autonomously in one of three phases: Initial, Online, and Offline. As represented in Figure 111 in the Initial Phase the service is in an Unknown state. How this phase is entered and exited is implementation and largely device dependent. From the Initial phase, the Service can transition only to the Offline Phase. In the Offline Phase the Service can be in the Down or Testing states. Although there may be access to the Service via the Console Subunit in this phase, such access does not include creating or processing Jobs; there is no remote access. Although transitions from Online phase states to the Down state are defined in this model, the manner and modes of testing and therefore the paths to and from the Testing phase are implementation dependent.

The basic Job-handling functions of the Service are done in the Online phase, which includes Idle, Processing and Stopped. The following paragraphs describe the nature of these states. Service States as status Elements, and the operations, events and conditions involved in changing states are described in more detail in section 7.2.

Note that the Resource Service does not deal with Jobs and Documents but rather with Resources. Therefore, although it goes through similar states, the sequence of operations is linked to Client Resource activity rather than Job activity.

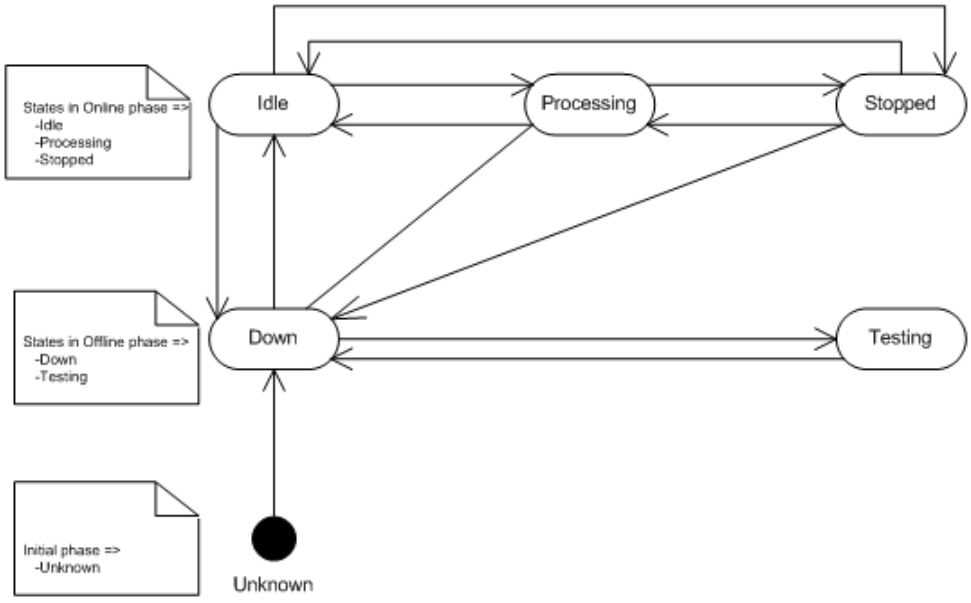

Figure 111 Service State Model

7.1.1 Initialization and Startup

On creation, an MFD Service enters its Initial phase during which all its service Elements and connected Subunits are initialized. This phase may include test of the configured Subunit(s) and self-testing of the Service. After successful initialization, the Service performs a start-up operation that brings it to the Online phase. Once in this phase, it authenticates and registers its service with an on-line service directory or otherwise announces its service to the network domain in which it resides. The Service is then in the "Idle" state, it is ready for service discovery and it can accept service requests from Clients.

The Service generally accepts new requests as long as it is in one of the three Online phase states: Idle, Processing or Stopped. However, performing an administrative Disable operation while in any state will stop the Service from accepting a CreateJob requests to start a new Job. (In the case of the Resource Service, a Disable operation will stop acceptance of any Resource "put" or "get" request.) Performing an Enable operation in any Online state while the Service is disabled will re-enable the Service to accept the disabled requests.

7.1.2 Job Creation

A user initiates a Job by submitting a CreateJob request through a local (via the MFD user interface) or remote (via local network or Internet) Client. The "intent" of the user in creating the Job, that is the processing and handling instructions, may be identified in a Job Template submitted with the CreateJob request, or aspects may be left to be determined by Service defaults. When the Job is created, the Service also creates a Job Ticket, a data object containing Job description and Job and Document processing instructions derived from the Job Template information. The Job Ticket remains bound to the Job until the Job is eventually deleted from the Service.

The Job Template has the form of a Job Ticket, but is unbound to any Job or Service. The Job Template for a specific Job can be created in several ways. (See Job and Job Ticket Life Cycle, section 2.6.)

The Service Capabilities of the selected Service can be retrieved to provide the set of supported features and values to allow creating or modifying a template so that it is appropriate to the service accepting the Job request. Once the Job Template fully embodies the Intent, it can be bound to a CreateJob request to send to the Service.

The Default Job Ticket is an Element of the Service. If a CreateJob request does not include a Job Template or if the Job Template did not include some required information, the applicable processing directive defaults to the Default Job Ticket values. The values from the Default Job Ticket are not copied to the Job's Job Ticket rather the combined Elements from the user supplied Job Ticket and the applied values from the Default Job Ticket are used. The Default Job Ticket can be administratively modified to control default behaviors for the associated Service.

The Service places created Jobs in the Active Jobs queue. Created Jobs are in the Pending state until they are advanced to the Processing State. A Pending Job may be held to delay scheduling for processing by a Job Hold Until Time Element in the Job's Ticket to allow time for user to walk up to the printer or scanner Subunit to handle the hardcopy Document; a Pending Job may also be held by an administrator Hold Job operation. Held Jobs are released by a Job Hold Until Time timeout or a administrator Release Job operation. Jobs that are not in a "held" state are scheduled for processing immediately or when a StartJob event is signaled, based on Job priority relative to other Jobs in the Active Jobs queue.

7.1.3 Job Processing

When a Job is released for scheduling and reaches the top of Active Jobs queue, it enters the Job Processing state. If the Service is not already in the Service Processing state, it will now enter that state. The Service manages and processes the Job according to the instructions in the Job Ticket,

supplemented where necessary by information in the Default Job Ticket. If a processing instruction in the Job Ticket cannot be supported, the Service either returns an error or substitutes the instruction with another that best matches the user's intent. When Job processing is complete, the actual values used for processing the Job are captured in the Job Receipt, the Job is in the Completed state and it is entered into the Job History queue. The Service either returns to Idle or continues in the ServiceProcessing state with a subsequent Job.

7.1.4 Service Pause, Service Shutdown and Abnormal Job Termination

During Job processing, an administrative "PauseService"or "PauseServiceAfterCurrentJob" operation will cause the Service to enter the "Stopped" state. This prevents further scheduling of Jobs for that Service. Depending on implementation, the Service transition to Stopped may be delayed to allow the processing of the current Job to be completed or suspended in a controlled way. Jobs that are in the Job Processing state when the Service is Stopped remain in the Active Jobs Queue but go into the ProcessingStopped Job state. A "PauseService" operation allows a user to interrupt the scheduled Jobs to submit and process an urgent Job or a Job for another service. A ServiceResume operation returns the Service to the Processing state and allows Job scheduling and processing to resume. Note that, if the Service Pause/Resume operations are to be used to allow a Job processing to be interrupted to allow another Job to be processed by the same service, then the inserted Job must have higher scheduling priority than a ProcessingStopped Job and the Service must readjust its schedule on receiving the Resume command.

When there are critical conditions impacting serviceability during "Idle" or "Processing" state, either a Critical Fault event is generated or an Administrative PauseService is performed to bring the service to the Stopped state. From there the condition can be fixed by user's intervention. Then either the Service generates a EndCritical event (removal of critical condition) or an administrator performs a Resume operation to bring the Service back to the "Idle" or "Processing" state. If the Service requires a ShutdownService operation before service can be resumed or for testing, a RestartService must be issued to bring the Service to the On-Line phase again.

Any Job in the Active Jobs queue can be canceled via a CancelJob operation by an authorized user, a. The Job is then in the Cancelled state and entered into the Job History queue. Jobs that are terminated because of a Service fault are in the Aborted state and are also entered into the Job History queue.

7.2 Service, Job and Document States

The values of the Service, Jobs and Document states are critical Elements in the Service, Job and Document models. Moreover, understanding what these states are and what is involved in transitioning from one state to another is also critical in understanding the operations, processes and conditions that control the ability of the Service to execute a Job and process a Document.

7.2.1 Service States, Conditions and State Transitions

The states of a Service are identified below and in Figure 111. The states and the events (including client requests) that cause state transitions are represented Figure 112. Service State and condition transitions as a result of Operations and Events are summarized in Table 74, Table 75 and Table 76.

7.2.1.1 Service States

The modeling of a Service is based on the following six states.

> **Unknown**: This is a state in the initial phase when a Service has been just created. There is no outside interface with the Service. Within the MFD, the Service will be initialized and will transition to the Down state.

Down: The Service instance exists and has been initialized. The service cannot accept any client operation request other than Start or Restart (which brings the Service to the Idle state, unless some error condition exists). The service can also be put into a Test state via some implementation-dependent path that initiates testing.

Testing: An implementation-dependent state in which normal processing is not possible but the Service is operating in a test mode.

Idle: The Service is up and able to accept client requests, but is not currently processing a Job. There may be no unterminated Jobs, or Jobs may be in a PendingHeld Job state.

Processing: At least one Job is in the Processing Job state.

Stopped: The Service is unable to process Jobs because it has received a Pause request or because an event signifying the existence of a critical condition has occurred.

7.2.1.2 Operations, Events and Conditions

Figure 112 illustrates the Service state transitions resulting from Operations and Events. The following notations are used.

Initial Caps = state; e.g., Idle
Initial caps followed by argument parenthesis = operation; e.g., Startup
Lower case = action; e.g., schedule, enable, etc
C. prefix = a condition; e.g., C.isAcceptingJobs means the isAcceptingJobs condition exists or is affirmative
E. prefix = an event; e.g., E.endJob means "Job completed"
~ = logical NOT applied to condition (e.g., ~C.paused means the Pause condition does not exist or "not paused")

Service Conditions and their initiating and terminating events are listed in Table 74. In general, Conditions may exist in several states and are not necessarily associated with state changes. Administrative Operations that cause Service State changes are identified in Table 75. Conditions and Events that characterize Service operation are identified in Table 76. Note that although no specific Test operation or events are defined in this specification, the Test state is included to indicate that the transition to and from this Service state is made in a specific manner.

Table 74 Conditions and Bounding Events

Condition	Initiating Event	Initiating Cause	Terminating Event	Terminating Cause
C.testing	E.testing	Implementation dependent	E.testingCleared	Implementation dependent
C.warning	E.warning	Some non critical equipment problem	E.warningCleared	Clearing non critical equipment problem
C.critical	E.critical	A critical equipment problem	E.criticalCleared	Clearing critical equipment problem
C.JobProcessing	E.startJob	A Created Job is scheduled	E.endJob	All scheduled Jobs advanced to Terminating state (Completed, Canceled, Aborted)
C.isAcceptingJobs	E.enable	Enable	E.disable	Disable
C.Pause	E.pause	Pause	E.resume	Resume

Note: C.isAcceptingJobs represents the ability of the service to create new Jobs (i.e., CreateJob will fail if ~C.isAcceptingJobs). Other operations are not affected by this condition.

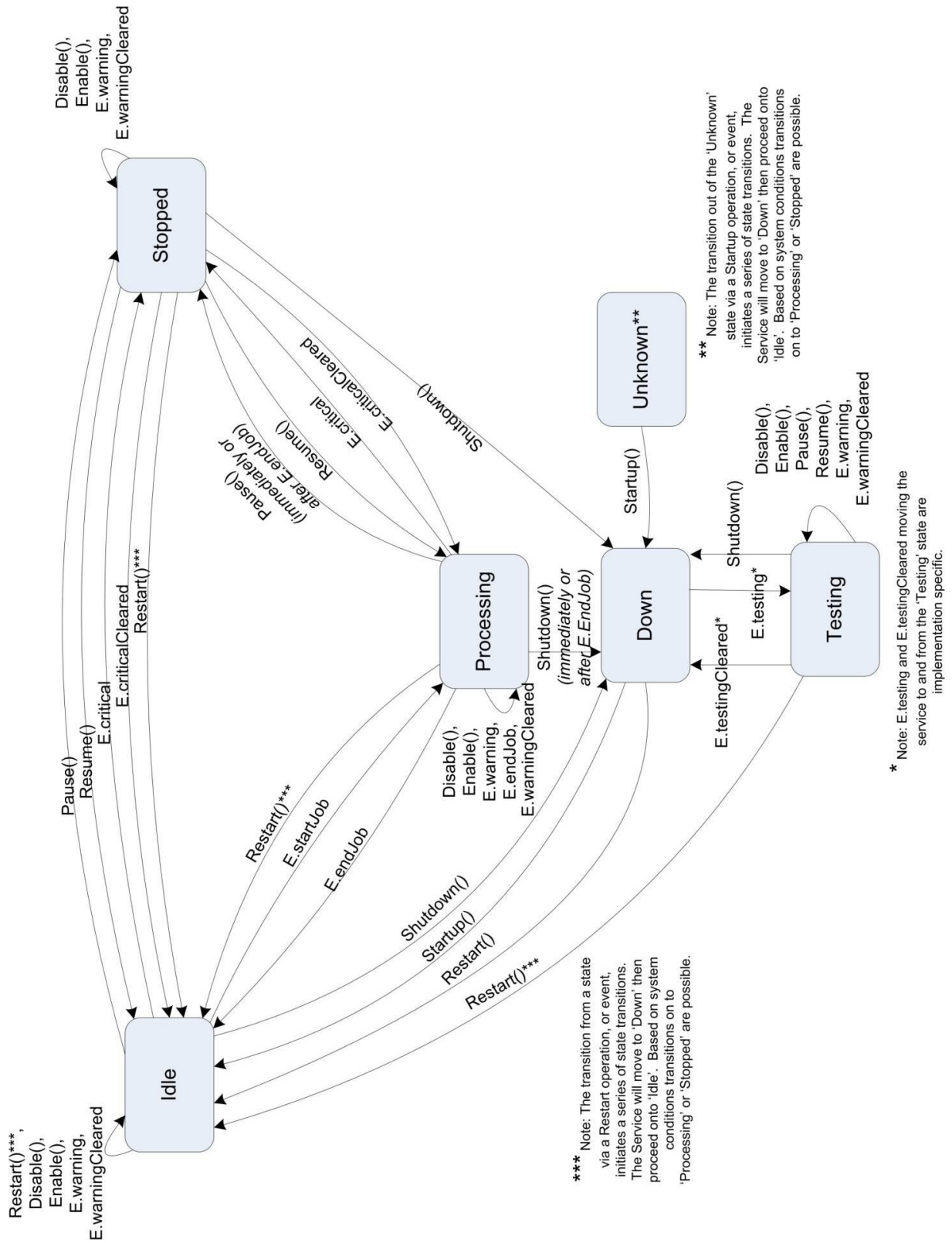

Figure 112 Service State Transitions and Causal Events

7.2.1.3 Service State Transition by Operations

Table 75 summarizes the effect of the primary Administrator operations on the Service state, depending upon the Service state when the operation request is received. The entry “error” indicates that sending in indicated operation when the Service is in the indicated state is invalid and an error response is returned to the requestor.

Table 75 ServiceState Change by Operations

	State				
	Down	**Testing**	**Idle**	**Processing**	**Stopped**
Operation	**Action (end state, condition)**	**Action (end state, condition)**	**Action (end state, condition)**	**Action (end state, condition)**	**Action (end state, condition)**
PauseService (Note 2)	error	pause (Testing, C.Pause)	pause (Stopped, C.Pause)	pause (Stopped, C.Pause)	pause (Stopped,C. Pause)
ResumeService	error	resume (Testing, ~C.Pause)	resume (Idle, ~C.Pause)	resume (Processing, ~C.Pause)	resume (Idle, ~C.Pause)
RestartService (Note 1)	restart (Idle)	restart (Idle)	restart (Idle)	restart (Idle)	restart (Idle)
ShutdownService (Note 2)	error	shutdown (Down)	shutdown (Down)	shutdown (Down)	shutdown (Down)
StartupService (Note 1)	restart (Idle)	error	error	error	error
test (Note 3)	test (Testing)	test (Testing)	error	error	error

Notes

1. StartupService is normally sent when the Service is first created. It prompts an E.startup event that initializes the Service and takes it through the Down state to Idle, assuming that there are no inhibiting conditions.
2. RestartService can be sent with the Service in any state. It prompts a restart which initializes the Service and takes it through the Down state to Idle, assuming that there are no inhibiting conditions.
3. StartupService sent when the Service is in the Down state has the same effect as a RestartService. StartupService is illegal in any other state.
4. PauseService and ShutdownService. Depending upon implementation:
5. A PauseService or ShutdownService received when the Service is in the Testing, Idle or Stopped states forces the Service immediately to the indicated state (Stopped and Down respectively).
6. ShutdownService when the Service is in the Processing state may initiate an E.shutdown event, leaving the Service in the Processing state while the Job in process is completed or otherwise terminated. The E.shutdown event will eventually cause the Service to go to the Down state.
7. PauseService when the Service is in the Processing state may initiate an pause event, leaving the Service in the Processing state while the Job in process is completed or otherwise terminated. The pause event will eventually cause the Service to go to the Stopped state.

7.2.1.4 Service State Transition and Condition Change by Events

Table 76 summarizes the Service state and condition change as a result certain events, depending upon the Service state when the event occurred. The table should be read from top to bottom. Consecutive entries of the same event are differentiated by differing associated conditions. The last entry in the series is equivalent to the event with a condition other than the ones already covered.

An “error” entry indicates an invalid circumstance and that an error message should be logged indicating an implementation error.

Table 76 Service State Change by Events

Input	State				
	Down	**Testing**	**Idle**	**Processing**	**Stopped**
Event (Condition)	**Condition (end state)**	**Condition (end state)**	**Condition (end state)**	**Condition (endstate)**	**Condition (end state)**
E.critical	C.critical	C.critical	C.critical (Stopped)	C.critical (Stopped)	C.critical
E.criticalCleared (i.e., last C.Critical)	~C.critical	~C.critical	error	error	~C.critical (Idle or Processing)
E.endJob (C.shutdown)	Error	shutdown (Down)	error	shutdown (Down)	error
E.endJob (C.paused)	error	C.paused	error	C.paused (Stopped)	error
E.endJob	error	---	error	schedule (Idle or Processing)	error
E. Warning	C.warning	C.warning	C.warning	C.warning	C.warning
E. warningCleared (i.e., last C.warning)	~C.warning	~C.warning	~C.warning	~C.warning	~C.warning
E.startup (Note 1)	restart (Idle)	error	error	error	error
E.startJob (C.paused)	error	error	error	error	error
E.startJob	error	schedule	schedule (Processing)	schedule	error
E.testing	(Testing)		error	error	error
E.testingCleared	error	(Down)	error	error	error

Notes

1. StartupService is normally sent when the Service is first created. It prompts an E.startup event that initializes the Service and takes it through the Down state to Idle, assuming that there are no inhibiting conditions.
2. RestartService can be sent with the Service in any state. It prompts a restart which initializes the Service and takes it through the Down state to Idle, assuming that there are no inhibiting conditions.
3. StartupService sent when the Service is in the Down state has the same effect as a RestartService. StartupService is illegal in any other state.
4. PauseService and ShutdownService. Depending upon implementation:
5. A PauseService or ShutdownService received when the Service is in the Testing, Idle or Stopped states forces the Service immediately to the indicated state (Stopped and Down respectively).
6. ShutdownService when the Service is in the Processing state may initiate an E.shutdown event, leaving the Service in the Processing state while the Job in process is completed or otherwise terminated. The E.shutdown event will eventually cause the Service to go to the Down state.
7. PauseService when the Service is in the Processing state may initiate an pause event, leaving the Service in the Processing state while the Job in process is completed or otherwise terminated. The pause event will eventually cause the Service to go to the Stopped state.

7.2.2 Job States and State Transitions

Jobs are created by a Service in response to a CreateJob request from a client. The Job will transition through several states under the control of the Service, and may be forced into several states by client Operations or equipment conditions. The possible Job states are listed below and the Job state transitions are diagramed in Figure 113. [RFC2911]

Note that a “saved Job” is not in a distinct state. It is a Completed Job with certain processing instructions. If a saved Job is recalled a ResubmitJob operation, a new Job is created and submitted with an implicit CreateJob request. The “saved Job” remains as a Completed Job.

Aborted the Job was halted due to an error.

Canceled the Job was halted by a CancelJob operation.

Completed the Job has been successfully completed.

Pending the Job has been accepted by the system and is awaiting system resources before it can start processing. It is in a queue and is scheduled for processing.

PendingHeld the Job is not a candidate for processing for any number of reasons and will return to the Pending state when the reasons are no longer present.

Processing the Job is being processed by the Service according to the Job Ticket instructions.

ProcessingStopped the Job is sent to the ProcessingStopped state when it is in the Processing state and the Service has been sent to the Stopped state. The Service may go to the Stopped state because of a PauseService operation, a Critical Event, or because the Service is awaiting resources necessary to continue processing.

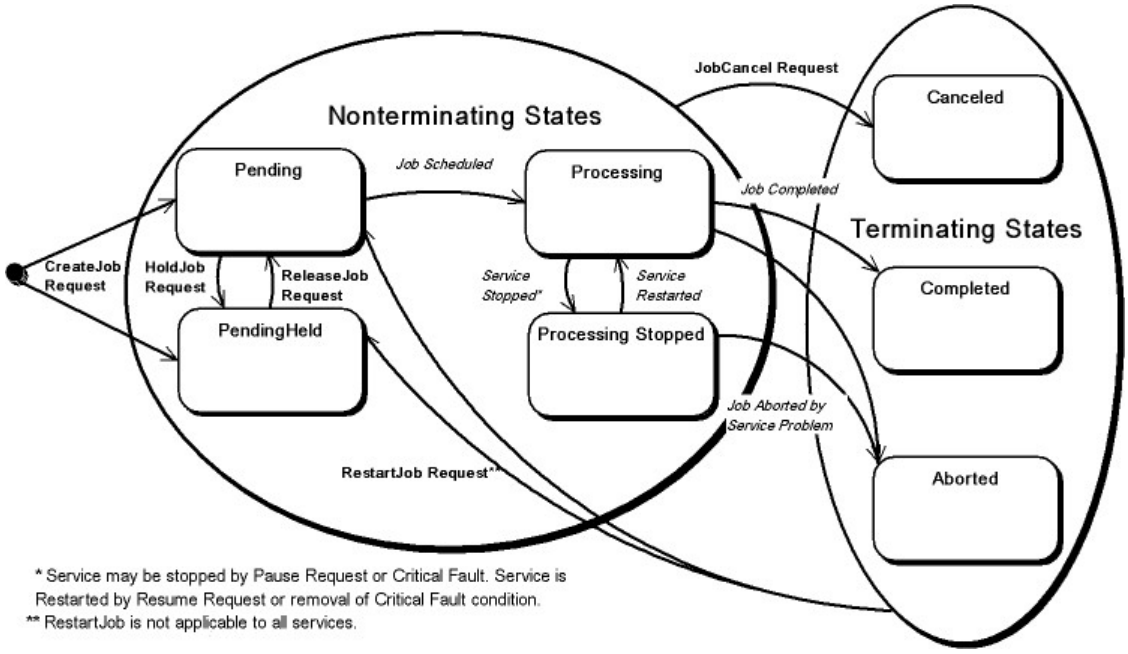

Figure 113 Job State Transition

7.2.3 Document States and State Transitions

Document states and state transitions largely parallel the states and state transitions of the Job that was created to process the Document. The states are listed below and the state transitions are diagrammed in Figure 114. [PWG5100.5].

Aborted The Document has been aborted by the system, usually while the Document was in the 'Processing' state and the Scan Service has completed aborting the Document and all Document status Elements have reached their final values for the Document. While the Scan Service is aborting the Document, the Document remains in its current state, but the Document's "DocumentStateReasons" Element SHOULD contain the 'ProcessingToStopPoint' and 'AbortedBySystem' values. When the Document moves to the Aborted state, the 'ProcessingToStopPoint' value, if present, MUST be removed, but the 'AbortedBySystem' value, if present, MUST remain.

Canceled The Document has been canceled as a result of the Scan Job being cancelled by a CancelScanJob" operation prior to the Document reaching a 'Completed' or 'Aborted' state. While the Scan Service is canceling the Document, the Document remains in its current state, but the Document's "DocumentStateReasons" Element SHOULD contain the 'ProcessingToStopPoint' value and one of the 'CanceledByUser', 'CanceledByOperator', or 'CanceledAtDevice' values. When the Document moves to the 'canceled' state, the 'ProcessingToStopPoint' value, if present, MUST be removed, but the 'CanceledByXxx', if present, MUST remain.

Completed The Document has completed successfully or with warnings or errors after processing and all of the Document's Digital Documents have been sent to their Destination and all Document status Elements have reached their final values for the Document. The Document's "DocumentStateReasons" Element SHOULD contain one of: 'CompletedSuccessfully', 'CompletedWithWarnings', or 'CompletedWithErrors' values.

Pending The Document has not started to be processed at all.

Processing The Job has begun using, or is attempting to use, one or more purely software processes that are analyzing, creating, or interpreting the Hard Copy Document or Digital Document. The Job has begun using, or is attempting to use, one or more hardware devices that are analyzing, creating, or interpreting the Hard Copy Document or Digital Document. The Digital Document is ready for sending to its Destination, but the Output Channel is not yet transferring it, either because the Digital Document hasn't reached the Output Channel or because the Document is queued in the Output Channel or some other spooler, awaiting the Output Channel to transfer it. The 'processing' state for the Document indicates that the Document has begun to be processed. Even if the Job stops being processed, the Document remains in the 'processing' state until it moves to one of the three terminal states. Implementations MAY include additional values in the Document's "DocumentStateReasons" Element to indicate the progress of the Document, such as adding the 'Scanning' value to indicate when the Scanner device is actually acquiring the Image from the Scan Region of the Hard Copy Document.

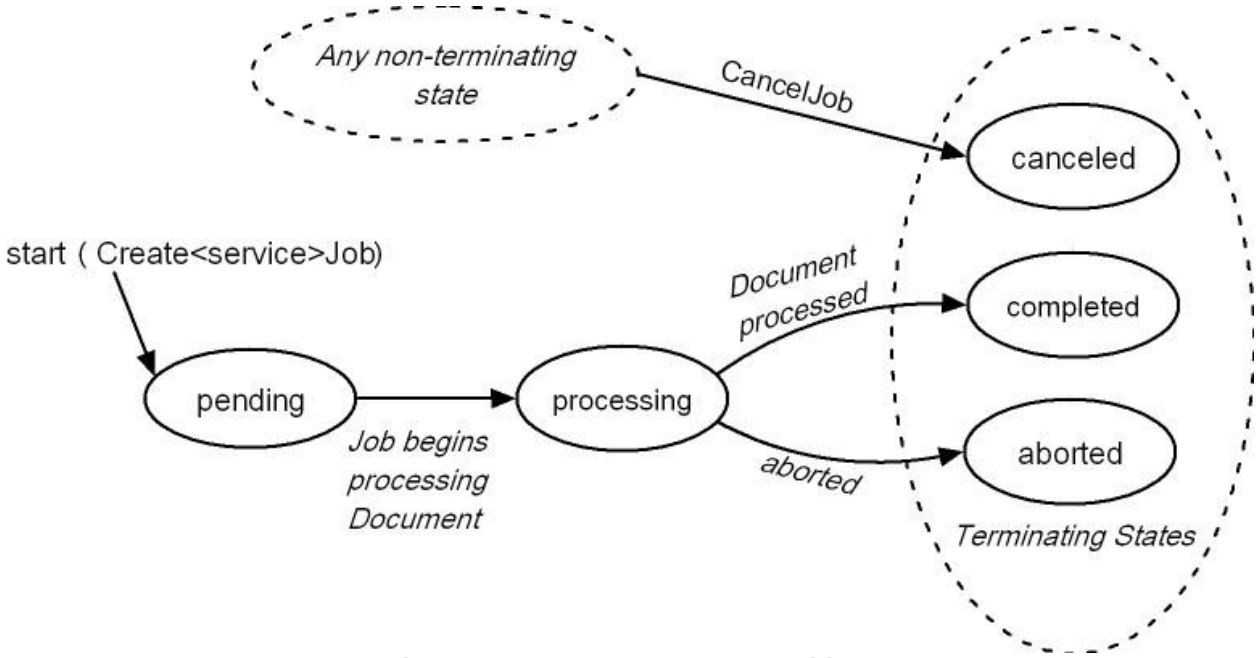

Figure 114 Document State Transition

7.3 Service Operations

This section defines normative semantics of Service operations and the behavior in response to these operation requests by a Service conformant to the MFD model.

A User makes a Service request by interacting through a local Client (via the MFD console) or through a remote Client via its software application user interface. Each Service provides the same set of service interfaces for the co-located local Client or a Remote Client. The Client can operate via a local interface, a local area network, or the Internet.

The individual Service specifications identify all operations applicable to that Service, some of which may be unique to that Service. The MFD Operations described in this section are common to more than one Service, with the "<service>" component in the operation name identifying the specific Service to which the operation request is addressed.

All operations consist of a Request issued by a client and a Response returned by the Service. All Requests are sent to the target Service except Startup<service>Service, which is sent to the MFD System. Some responses may just indicate that the request was or was not honored, perhaps with explanation messages; others will contain requested or related information, perhaps with additional explanatory "reasons" information.

The MFP operations applicable to two or more Services are listed in Table 77 along with references to their antecedent Printer operations. The operations are, for the most part, derived from IPP Print Service operations defined in RFC2911 [RFC2911], RFC3380[RFC3380] and RFC3998 [3998] and further discussed in the PWG IPP 2.0 specification [PWG5100.10] and the IPP Job and Printer Extensions-Set 2 specification [PWG5100.11]. The print specific context has been extended to MFD Services. Several print operations have been omitted as inapplicable to MFD Services as a whole, while a few operations have been added.

Table 77 MFD Common Operations and Antecedents

MDF Operation	Antecedent IPP Operation	Antecedent Reference		Access*
Add<service>HardcopyDocument	Send-URI**	[RFC2911]		User
Cancel<service>Document	Cancel-Document	[PWG5100.5]	4.5	User
Cancel<service>Job	Cancel-Job	[RFC2911]	3.3.3	User
Cancel<service>Jobs	Cancel-Jobs	[PWG5100.11]	5.1	Admin
CancelCurrent<service>Job	Cancel-Current-Job	[RFC3998]	4.2	User
CancelMy<service>Jobs	Cancel-My-Jobs	[PWG5100.11]	5.2	User
Close<service>Job	Close-Job	[PWG5100.11]	5.3	User
Create<service>Job	Create-Job Print-Job	[RFC2911]	3.1.4 3.2.1	User
Disable<service>Service	Disable-Printer	[RFC3998]	3.1.1	Admin
Enable<service>Service	Enable-Printer	[RFC3998]	3.1.2	Admin
Get<service>DocumentElements	Get-Document-Attributes	[PWG5100.5]	4.3	User
Get<service>Documents	Get-Documents	[PWG5100.5]	3.3	User
Get<service>JobElements	Get-Job-Attributes	[RFC2911]	3.3.4	User
Get<service>Job History	Get-Jobs (which-Jobs Element = ‘completed’)	[RFC2911]	3.2.6	User
Get<service>ServiceElements	Get-Printer-Attributes, Get-Printer-Supported-Values	[RFC2911], [RFC3380]	3.2.5	User
GetActive<service>Jobs	Get-Jobs (which-Jobs Element = ‘not-completed’)	[RFC2911]	3.2.6	User
Hold<service>Job	Hold-Job	[RFC2911]	3.3.5	User
HoldNew<service>Jobs	Hold-New-Jobs	[RFC3998]	3.3.1	Admin
Pause<service>Service	Pause-Printer	[RFC2911]	3.2.7	Admin
Pause<service>ServiceAfterCurrentJob	Pause-Printer-After-Current-Job	[RFC3998]	3.2.1	Admin
Promote<service>Job	Promote-Job Schedule-Job-After	[RFC3998] [RFC3998]	4.4.1	Admin
Release<service>Job	Release-Job	[RFC2911]	3.3.6	User
ReleaseNew<service>Jobs	Release-Held-New-Jobs	[RFC3998]	3.3.2	Admin
Restart<service>Service	Restart-Printer	[RFC3998]	3.5.1	Admin
Resubmit<service>Job	Resubmit-Job	[PWG5100.11]	5.4	User
Resume<service>Job	Resume-Job	[RFC3998]	4.3.2	User
Resume<service>Service	Resume-Printer	[RFC2911]	3.2.8	Admin
Send<service>Document	Send-Document	[RFC2911]	3.3.1	User
Send<service>URI	Send-URI	[RFC2911]	3.3.2	User
Set<service>DocumentElements	Set-Document-Attributes	[PWG5100.5]	4.4	User
Set<service>JobElements	Set-Job-Attributes	[RFC3380]	4.2	User
Set<service>ServiceElements	Set-Printer-Attributes	[RFC3380]	4.1	Admin
Shutdown<service>Service	Shutdown-Printer	[RFC3998]	3.5.2	Admin
Startup<service>Service	Startup-Printer	[RFC3998]	3.5.3	Admin
SuspendCurrent<service>Job	Suspend-Current-Job	[RFC3998]	4.3.1	User

Notes:

* For operations where a User may set Job Elements or otherwise directly affect the state or parameters of a Job, the User must be either the Job owner of record or must be an Administrator or Operator. For operations where a User has access to Job or Service information (Get<service>JobElement, Get<service>JobHistory and Get<service>Jobs), site policy determines what information is made available to Users who are not Administrators or Operators and who do not own the Jobs about which the information is provided.

** Although the Add<service>HardcopyDocument operation is a “new” operation created to address the input to Services accepting Hardcopy versus Digital Documents, the purpose and form of the operation is parallel to the IPP Send-URL operation

The MFD operation definitions in this section are generic. Depending on the encoding used by the binding, the actual identification of the operation may be different. For example, IPP uses a numeric code. Further, depending on the addressing inherent in the transport, the operation requests might include an implicit rather than explicit identification of the Service. For example, IPP operations coming on the TCP port 631 are inherently Print Service operations.

The MFD Operation definitions are divided between basic or User (Job-oriented) operations and administrative operations. The basic MFD operations are listed in Table 78. These operations are concerned primarily with creating, monitoring, modifying and canceling Jobs and Job-related Elements. Basic operations are available to Users including Administrators and Operators, although any operation affecting a Job or Document is restricted to the Job Owner or to an Administrator or Operator. Identification and authentication of the User as Job Owner depends upon the Service and binding, as well as the specific implementation. For example, the Copy Service may consider whoever is present at the machine to be the Job Owner.

Site Policy may cause a Service to restrict information provided to a User who is not an Administrator or Operator. Administrative Operations, accessible only to Administrators (and Operators), are concerned primarily with managing the Service and are listed in Table 79. Note that for some Services where the User is present at the device (such as Copy), certain operations may consider any User that is present at the implementing device as having Administrator access

.

Table 78 Basic MFD Interface Requests and Responses

Operation	Request Parameters (Notes 2)	Response Parameters (Note 3)	Note
Add<service>HardcopyDocument	InputSource, JobId, Document Ticket(optional), ElementsNaturalLanguage(optional), LastDocument(optional), RequestingUserName	DocumentNumber, UnsupportedElements(optional)	
Cancel<service>Document	DocumentNumber, ElementsNaturalLanguage(optional), JobId, Message (optional) RequestingUserName		
Cancel<service>Job	ElementsNaturalLanguage(optional), JobId, Message (optional) RequestingUserName		
CancelCurrent<service>Job	ElementsNaturalLanguage(optional), JobId(optional), Message (optional) RequestingUserName		
CancelMy<service>Jobs	JobIds (optional), Message (optional), ElementsNaturalLanguage(optional), RequestingUserName	JobIds (optional)	1
Close<service>Job	JobId, RequestingUserName		
Create<service>Job	ElementsNaturalLanguage(optional), Job Ticket (optional) RequestingUserName	JobId, UnsupportedElements(optional)	
GetActive<service>Jobs	ElementsNaturalLanguageRequested(optional), Limit(optional) RequestingUserName	ElementsNaturalLanguage(optional) JobSummaries (includes JobID, JobName, JobOriginatingUserName, JobState and perhaps JobStateReasons)(optional)	
Get<service>DocumentElements	Document Number, ElementsNaturalLanguageRequested(optional), JobId, RequestingUserName	DocumentElements(optional), ElementsNaturalLanguage(optional)	
Get<service>Documents	ElementsNaturalLanguageRequested(optional), JobId, RequestingUserName	Documents(list of DocumentSummaries)(optional), ElementsNaturalLanguage(optional) JobID, JobName	
Get<service>JobElements	ElementsNaturalLanguageRequested(optional), JobId, RequestedElements (JobReceipt, JobStatus, or Job Ticket.)(optional) RequestingUserName	JobElements, ElementsNaturalLanguage(optional)	

Operation	Request Parameters (Notes 2)	Response Parameters (Note 3)	Note
Get<service>Job History	ElementsNaturalLanguageRequested(optional), Limit(optional) RequestingUserName	ElementsNaturalLanguage(optional) JobSummaries (includes JobID, JobName, JobOriginatingUserName, JobState and perhaps JobStateReasons	
Get<service>ServiceElements	ElementsNaturalLanguageRequested(optional), RequestedElements (Service Capabilities, ServiceConfiguration, ServiceDescription, ServiceStatus or DefaultJob Ticket.)(optional) RequestingUserName	ElementsNaturalLanguage(optional) ServiceElements(optional)	
Hold<service>Job	ElementsNaturalLanguageRequested(optional), JobHoldUntil or JobHoldUntilTime, JobId, Message(optional), RequestingUserName		
Release<service>Job	ElementsNaturalLanguageRequested(optional), JobId, Message(optional), RequestingUserName		
Resubmit<service>Job	ElementsNaturalLanguageRequested(optional), JobId, Job Ticket (optional) RequestingUserName	JobId, UnsupportedElements(optional)	
Resume<service>Job	ElementsNaturalLanguageRequested(optional), JobId, Message(optional)RequestingUserName		
Send<service>Document	ElementsNaturalLanguageRequested(optional), Document Ticket (optional) JobId, LastDocument(optional), RequestingUserName, DocumentData	DocumentNumber, UnsupportedElements(optional)	
Send<service>Uri	DocumentUri, ElementsNaturalLanguageRequested(optional), Document Ticket (optional) JobId, LastDocument(optional), RequestingUserName	DocumentNumber, UnsupportedElements(optional)	
Set<service>DocumentElements	DocumentNumber, ElementsNaturalLanguage(optional), SocumentTicket, JobId, Message(optional), RequestingUserName	UnsupportedElements(optional)	
Set<service>JobElements	ElementsNaturalLanguage(optional), Job Ticket, JobId, Message(optional), RequestingUserName	UnsupportedElements(optional)	
SuspendCurrent<service>Job	ElementsNaturalLanguage(optional), JobId(optional), Message(optional), RequestingUserName		
Validate<service>Document Ticket	ElementsNaturalLanguageRequested(optional), Document Ticket RequestingUserName	UnsupportedElements(optional)	
Validate <service>Job Ticket	ElementsNaturalLanguage(optional), Job Ticket, RequestingUserName	UnsupportedElements(optional)	

Notes:

Note 1: Response includes identified but un-cancellable Jobs

Note 2: The RequestingUserName, is used by the Service to determine whether the requestor is an Administrator, Operator or the Job Owner and is therefore authorized to make the request. Some implementations may require further authentication of the requestor's identity. If the requestor is not determined to have access, the Service MUST reject the request.

Note 3: All responses must include correlation to request and whether request was successful or failed.

7.3.1 Basic Service Operations

The common Basic operations are listed in Table 78; they are concerned with creating and controlling Jobs and Documents within Jobs. The Operations include those by which a client gets Service Elements to allow selection of Services and formulation of Job Tickets. Some of these operations do affect the state of a Job. However, none of these operations directly affect the state or configuration of the Service except to the extent that creating or canceling a Job may initiate a sequence that affects the Service.

7.3.1.1 Add<service>HardcopyDocument

The Add<service>HardcopyDocument operation allows a client to prepare a Service to accept a Hardcopy Document via a scanner Subunit and to add it to an identified Job. It is analogous to the Send<service>Document and Send<service>Uri operations except that it is applicable to Services for which input Documents are obtained by a scan of a region of a media sheet side, such as FaxOut and EmailOut.

The Service MUST reject this request and send an appropriate message if:

1. The requestor is not the owner of the identified Job, or is not an Administrator or Operator;
2. The Service has already closed inputs to the identified Job, or
3. The Job is not found.

Otherwise, provided the request is properly constructed, complete and references valid objects, the Service MUST accept the request, MUST close the Job if the LastDocument Element is asserted, MUST be prepared to add Document Data from the identified input to the identified Job, and MUST respond to the request.

7.3.1.2 Cancel<service>Document

The Cancel<service>Document operation allows a client to cancel a specified Document in a specified Job of the specified Service any time from when the time the Document is created up to, but not including, the time that the Document is Completed, Canceled or Aborted. Because a Document might already be in Processing by the time a Cancel<service>Document request is received, some portion of the Document processing might be completed before the Document is actually canceled.

The Cancel<service>Document operation does not remove the Document from the Job or the Service, but does set the specified Document's Document State Document Status Element to Canceled and the Document's Document State Reasons Element to an appropriate value. If the Job containing the Document is again submitted using Resubmit<service>Job, the canceled Document is also submitted for processing. Thus Cancel<service>Document has the same semantics as Cancel<service>Job which cancels only the processing of the Job but does not delete the Job object itself.

The Cancel<service>Document operation does not affect the states of any of the other Documents in the Job. If the Job is in the Processing state and there are more Documents to be processed, the Service does continue to process the un-canceled Documents. If there are no further Documents to process, the Job is advanced to the Completed state.

The Service MUST reject the operation and return an appropriate response message if the operation requestor is not either the Job owner or a Service or System operator or administrator. Otherwise the Service MUST accept or reject the Cancel<service>Document request based on the Document's current state and, if the request is accepted, the Service MUST transition the Document to the indicated new state as follows:

Once a "success" response has been sent, the implementation guarantees that the Document will eventually end up in the Canceled state. Between the time that the Cancel<service>Document request is accepted and when the Document enters the Canceled Document-state, the DocumentStateReasons Element MUST contain a value which indicates to any later query that, although the Document might still be Processing, it will eventually end up in the Canceled state.

7.3.1.3 Cancel<service>Job

The Cancel<service>Job operation changes the state of the identified Job to Canceled, provided that the Job is not already in or in a mode leading directly to a termination state. (i.e., Completed, Canceled, or

Aborted.) Because a Job might already be active by the time a Cancel<service>Job is received, a portion of the Job may be done before the Job is actually terminated.

The Service MUST accept or reject the request based on the Job's current state. If the request is accepted, the Job state is transitioned to Canceled and the Service will issue a success response. See the transition diagram under Job State (Figure 113). If the implementation requires some significant time to cancel a Job in the Processing or ProcessingStopped states, the Service MUST set the Job's JobStateReasons to a value indicating that the Job is transitioning to a Canceled state. If the Job already has a JobStateReasons indicating that it is transitioning to a Canceled state, then the Service MUST reject a Cancel<service>Job operation

7.3.1.4 CancelCurrent<service>Job

The CancelCurrent<service>Job operation allows a client to cause the Service to terminate processing on the currently processing Job and to move that Job to the Canceled state. As with any other Basic operation directly affecting a Job, this operation is accepted by the Service only if the originator is the Owner of the affected Job(s) or is an Administrator or Operator.

There is the potential that the current Job may have changed between the time a client requests this operation and the time the Service implements it. Therefore, if the intent is to cancel a particular Job the Client MAY include an optional JobId parameter in the request.

1. If the JobId is included in the request and that Job is currently in the Processing or ProcessingStopped state and the operation requestor has access rights to that Job, the Service MUST accept the request and cancel the Job.
2. If no JobId is included in the request and the operation requestor has access rights to the Job currently in the Processing or ProcessingStopped state, the Service MUST accept the request and cancel that Job.
3. If more than one Job is in the Processing or ProcessingStopped state, all currently processing Jobs to which the request originator has access MUST be canceled unless the operation included the optional JobId, in which case only the identified Job is canceled.
4. If the JobId is included in the request and that Job is not currently in the Processing or ProcessingStopped state; or if the requestor does not have access rights to the identified Job, the Service MUST reject the request and return the appropriate error code.
5. If there is no Job currently in the Processing or ProcessingStopped state or if the requestor does not have access rights to any Job that is in the Processing or ProcessingStopped state, the Service MUST reject the request and return the appropriate error code.

7.3.1.5 CancelMy<service>Jobs

The CancelMy<service>Jobs operation permits a user to cancel all of their own identified non-Terminated Jobs or, if no specific Jobs are identified in the request, to cancel all of their own non-Terminated Jobs in the Service. This operation works like the Cancel-Job operation except that the operation can apply to multiple Jobs. The client specifies the set of candidate Jobs to be canceled by supplying and/or omitting the JobIds. The Service MUST check the access rights of the requesting user against *all* of the candidate Jobs. If *any* of the candidate Jobs are not owned by the requesting user, the Service MUST NOT cancel any Jobs and MUST return the appropriate error status code along with the list of any JobIds that were specifically identified in the operation request but to which the User is not authorized access.

If this check succeeds, then (and only then) the Service MUST accept or reject the request based on the current state of each of the candidate Jobs and must transition each Job to the indicated new state as shown for the antecedent Cancel-My-Jobs operation in the Standard for Internet Printing Protocol (IPP): Job and Printer Extensions Set 2 [PWG5100.11]. If any of the candidate Jobs that were not already in a Terminating state cannot be canceled, the Service MUST NOT cancel any Jobs and MUST return the appropriate error status code along with the list of JobIds for those Jobs which were specifically identified in the operation request but could not be canceled. If the requested Jobs include some Jobs that are

already in a terminating state, this circumstance in itself MUST NOT interfere with the canceling of non-terminated candidate Jobs, but SHOULD result in the return of a warning message identifying the specifically identified Jobs that already were in a Terminating state.

7.3.1.6 Close<service>Job

The Close<service>Job operation allows a client to close Job inputs to those Services accepting Documents, even when the last Document input operation for the Job (Send<service>Document, Send<service>URI or Add<service>Document) did not include the LastDocument Element with a 'true' value. This Close<service>Job operation supersedes and, if supported by the Service, is preferable to the practice of using a Send<service>Document with no Document Data but with a LastDocument Element containing a 'true' value to close inputs.

The Service MUST reject this operation request if the target Job is not found or if the requestor is not the Job Owner or an Administrator. Otherwise, the Service MUST accept this operation request even if the target Job is already closed and regardless of JobState. Closing the Job MUST cause the Service to reject any subsequent Document input operation for the target Job, but MUST NOT affect the execution of any previously accepted Document input operation.

7.3.1.7 Create<service>Job

The Create<service>Job operation allows a Client to request creation of a Job in the Service. Upon creation, the Job is in Pending state and available for scheduling unless a Job Processing instruction prevents this. (e.g., JobHoldUntil puts it in PendingHeld state) The Create<service>Job operation MUST fail if the Service's IsAcceptingJobs Element value is 'false'.

Job Processing is done on one or more Documents. Unlike the antecedent IPP Print-Job operation, the MFD Create<service>Job may involve more than one Document. Depending upon the type of Service, the input may be a Hardcopy Document or a Digital Document. In either case, the source(s) of the input Document(s) as well as the destination(s) of the output Document(s) are identified in the Job Ticket submitted in the Create<service>Job Request,

Once a Job is created, Documents may be input as part of that Job by Send<service>Document, Send<service>URI or, for Services that accept hardcopy input, Add<Service>Document operations. In Service implementations that do not accept multiple Documents (i.e., MultipleDocumentJobsSupported = False), Document input is closed after one Document is accepted. In Service implementations that do accept multiple Documents (i.e., Multiple Document Jobs Supported = True), there may be multiple Send<service>Document, Send<service>URI or Add<Service>Document operations. There are two methods of indicating when all Documents have been sent:

1. issuing a Close<service>Document request
2. issuing a Send<service>Document, Send<service>URI or,
3. Add<Service>Document request with the LastDocument Element = True

To avoid a possible hang condition, Service implementations supporting multiple Document Jobs must also support the Multiple Operation Time Out Element that indicates the minimum number of seconds the Service will wait for the next Send or Add operation before taking some recovery action. If, for some reason, there is a longer period between Create<service>Job and valid Send or Add operations, or between sequential Send or Add operations, the Client MUST send Send or Add requests, even if they are empty, to reset the timeout. If there is a multiple operation timeout, the Service will take remedial action according to the value that Service has indicated in its Multiple Operation Timeout Action Element.

7.3.1.8 Get<Service>DocumentElements

The Get<Service>DocumentElements operation allows a Client to obtain detailed information about the specified Document within the specified Job. This operation is parallel to the Get<service>Job-Elements operation, but with the target and response Elements relating to a Document rather than a Job.

The Client requests specific groups of Elements (complex Elements) contained within the Document. The Document Data is not part of the Document and cannot be retrieved using this operation. However the location of the Document Data is available. The allowed values for Requested Elements are Document Receipt, Document Status and Document Ticket. Vendors may extend the allowed values.

The Service MUST return the Document Description Element values that a client supplied in the Document Creation operation (Create<service>Job, Send<service>Document or Send<service>URI) or provided in Set<service>DocumentElements operation a plus any additional Document Description Elements that the Service has generated, such as Document State. The Service MUST NOT return any Job level Elements that the Document inherits from the Job level but MUST return Document Elements specified at the Document level. It is NOT REQUIRED that a specific Document include all Elements belonging to a group (since some Elements are optional). However, it is REQUIRED that the Service support all these group names for the Document object.

7.3.1.9 Get<service>Documents

The Get<service>Documents operation allows a client to retrieve the list of Documents belonging to the identified Job. A Document summary containing a group of Document Element names with their values will be returned for each Document in the Job.

This operation is similar to the Get<service> and Get<service> operations except that it returns Elements from Documents rather than identified Jobs. As with the Get<service>DocumentElements operation, the Service MUST return only those Elements that are in the Document Ticket.

7.3.1.10 Get<service>JobElements

The Get<Service>JobElements operation allows a Client to obtain detailed information on the specified Job. Unlike the antecedent IPP Get-Job-Attributes operation, the Get<Service>JobElements request may not specify individual Elements. Rather, the Client requests specific groups of Elements contained within the Job. The allowed values for RequestedElements are Job Receipt, Job Status, or Job Ticket. Vendors may extend the allowed values.

The Service MUST reject this request if the requestor is not authorized access to the identified Job,

7.3.1.11 Get<service>Jobs

The Get<service>Jobs operation provides summary information on all Jobs that have reached a terminating state (i.e., Completed, Canceled Aborted). As such, it is similar to the antecedent Get-Jobs operation with the which-Jobs Element set to ‘completed’. Unlike Get-Jobs, Get<service>Jobs may not include a Requested Elements argument; rather, it always returns a Job Summary for each terminated Job including JobId, JobName, JobOriginatingUserName, JobState and perhaps JobStateReasons and other service specific information.

When the operation is exercised by a User that is not an Administrator, the Job summary may not include all of the summary information, depending upon site security policy.

7.3.1.12 Get<service>ServiceElements

The Get<service>ServiceElements operation allows a Client to obtain detailed information on the Elements and their values supported by the Service. Unlike the antecedent IPP Get-Printer-Attributes operation, the Get<Service>ServiceElements request may not specify individual Elements. Rather, the Client requests information on one or more specific group of Elements. The allowed values for Requested Elements are Service Capabilities, Service Configuration, Service Description, Service Status or DefaultJob Ticket. Vendors may extend the allowed values.

Some Services may accept an additional argument in a Get<service>ServiceElements request to further filter the response, much as the antecedent IPP Get-Printer-Attributes operation accepted the Document-Format Element. The individual Service specifications identify such arguments if any, their effect and whether support is mandatory.

In addition to the status message, the Service response includes the set of requested Element names and their values for all supported Elements. The response need not contain the requested Element names for any Elements not supported by the Service.

7.3.1.13 GetActive<service>Jobs

The GetActive<service>Jobs operation provides summary information on all Jobs in the Pending or Processing state. As such, it is equivalent to the antecedent Get-Jobs operation with the which-Jobs Element set to ‘not-completed’. Unlike the antecedent Get-Jobs operation, GetActive<service>Jobs may not include a RequestedElements argument; rather, it always returns a JobSummary for each Active Job with the summary including JobId, JobName, JobOriginatingUserName, JobState and perhaps JobStateReasons and other service specific information.

When the operation is exercised by a User that is not an Administrator or Operator, the Job summary may not include all of the summary information, depending upon site security policy.

7.3.1.14 Hold<service>Job

> NOTE: The antecedent Hold-Job operation and the associated Release-Job operation, as defined in IPP/1.1: Model and Semantics [RFC2911], have been deprecated in later IPP specifications in favor of using the antecedent Set-Job-Attributes operation [RFC3380] to set the Hold-Job-Until or Hold-Job-Until-Time attributes.

The Hold<service>Job operation allows a client acting for the Job Owner or an Administrator or Operator to hold a Pending Job in the queue so that it is not eligible for scheduling. The Job transitions as a result of a Hold<service>Job operation depend upon the current Job state, as indicated for the antecedent Hold-Job operation in paragraph 3.3.5 of IPP/1.1: Model and Semantics [RFC2911] The Hold<service>Job request can specify hold until a specific date-time (JobHoldUntilTime) or according to a keyword (JobHoldUntil), where the keyword can specify a period (such as “third-shift”) or be indefinite. A given Hold<service>Job request can specify only one hold condition. In the case of multiple Hold<service>Job requests, the last accepted request overrides the condition imposed by any previous Hold<service>Request.

The restraint imposed by a Hold<service>Job is removed by a Release<service>Job operation directed to the same Job. If a Service implementation supports Hold<service>Job, it must also support Release<service>Job and vice-versa.

If the HoldJob operation is supported, then the ReleaseJob operation MUST be supported, and vice-versa. The OPTIONAL JobHoldUntil or JobHoldUntilTime parameter allows a client to specify whether to hold the Job until a specified time, indefinitely or until a specified time period. The Service MUST accept or reject the request based on the Job's current state and transition the Job to the indicated new state as

follows. A HoldJob request is rejected when the identified Job is in the Processing or ProcessingStopped states.

7.3.1.15 Release<service>Job

> NOTE: The antecedent Release-Job operation and the associated Hold-Job operation, as defined in IPP/1.1: Model and Semantics [RFC2911], have been deprecated in later IPP specifications in favor of using the antecedent Set-Job-Attributes operation [RFC3380] to set Hold-Job-Until or Hold-Job-Until-Time attributes.

The Release<service>Job operation allows a client acting for the Job Owner or an Administrator or Operator to release a previously held Job from the PendingHeld state so that it is eligible for scheduling, provided that there is no other reason to keep the Job in the PendingHeld state. That is, the restraint imposed by a Hold<service>Job operation is removed by a Release<service>Job operation directed to the same Job. If a Service implementation supports Hold<service>Job, it must also support Release<service>Job and vice-versa.

The Job Transitions as a result of a Release<service>Job operation depend upon the current Job state, as indicated for the antecedent Release-Job operation in paragraph 3.3.6 of IPP/1.1: Model and Semantics [RFC2911]

7.3.1.16 Resubmit<service>Job

The Resubmit<service>Job operation allows a client acting for the Job Owner or an Administrator or Operator to resubmit a previously completed Job, but with the option of providing new Job Ticket information (other than input Document Data or input Document Data descriptive information.)

The Resubmit<service>Job operation is applicable only to a RetainedJob. A Retained Job is one which remains in the Service after it has been completed or canceled. This may be incidentally or because it is a saved Job, which is a Completed or Canceled Job with a JobSaveDispostion Element value that indicates that the Job, including Document Data if any, should not be deleted or aged-out after the Job is completed.

If a Resubmit<service>Job operation is accepted, the state of the retained Job is not changed; rather, a new Job is created from the identified retained Job and submitted with an implicit **CreateJob** request.

1. If the Resubmit<service>Job request contains a processing Element that was in the retained Job but with a different value, the value supplied in the Resubmit<service>Job operation MUST override the original value (if supported by the Service).
2. If the Resubmit<service>Job request contains a processing Element that was not in the retained Job, the Element with the value supplied with the Resubmit<service>Job operation MUST be applied (if supported by the Service)
3. For any processing Element in the original retained Job the value of which is not changed in the Resubmit<service>Job request, that Element and its value MUST be applied to newly created Job except that a JobSaveDispostion Element value indicating that the Job should be saved, and certain other Service-specific Element values, MUST NOT be copied but are applied to the new Job only if they are in the Resubmit<service>Job request.

The newly created Job is moved to the Pending or PendingHeld Job state with the same Element values as the original saved Job (except for the save Element). If any of the Documents in the saved Job were passed by reference (Send<service>URI or Send>service>URI), the Service MUST re-fetch the data, since the semantics of Restart<service>Job are to repeat all Job processing. The Service MUST assign new JobUri and JobId values to the newly created Job; the JobDescription Elements that accumulate Job progress, such as JobImpressionsCompleted, JobMediaSheetsCompleted, and JobKOctetsProcessed, MUST be an accurate record for the newly created Job.

The Service MUST accept or reject the Resubmit<service>Job Request based on the authority of the requester and the referenced Job's current state. The Requester must either be the Job owner or an operator or administrator of the Service. The target Job must be retained with a Completed or Canceled state.

7.3.1.17 Resume<service>Job

The Resume<service>Job operation allows a client acting for the Job Owner or an Administrator or Operator to resume the identified Job at the point where it was suspended. Provided that no other condition exists that forces the Job to the PendingStopped state, the Service moves the Job from the ProcessingStopped state to the Pending state and removes the JobSuspended value from the Job's StateReasons Element. If the identified Job is not in the ProcessingStopped state with the JobSuspended value in the Job's StateReasons Element, the Service MUST reject the request and return an appropriate status code, since the Job was not suspended.

If a Service supports Suspend<service>Job or SuspendCurrent<service>Job operations, it MUST support the Resume<service>Job operation, and vice-versa.

7.3.1.18 Send<service>Document

The Send<service>Document operation allows a client acting for the Job Owner or an Administrator or Operator to input a Digital Document to a Service as part of an already created Job. In response to the Create<service>Job, the Service will have returned the JobURI and the JobId. For each Document that the client desires to add to this Job, the client issues a Send<service>Document request which includes the JobId and contains the entire stream of Document Data for one Document.

If the Service supports this operation but does not support multiple Documents per Job, Document input is closed after the first Document is accepted and the Service MUST reject subsequent Send<service>Document requests associated with the same Job. Similarly, if the Service does support multiple Documents per Job, the Service MUST reject Send<service>Document requests associated with a given Job after inputs to that Job have been closed either a Close<service>Job operation or a previous Send<service>Document with a 'true' value for the LastDocument Element. Note that the Client may send and the Service must accept a Send<service>Document request with a 'true' value for the LastDocument Element to close input to that Job, even if that request includes no Document data.

See the Create<system>Job description for discussion of issues relating to excessive delay between multiple Send<service>Document requests.

The Service MUST reject a Send<service>Document request and send an appropriate message if:

1. The requestor is not the owner of the identified Job, or is not an Administrator or operator
2. The Service has already closed inputs to the identified Job,
3. The Document size, format and/or compression are not supported by the Service, or
4. The Job is not found.

Otherwise, the Service MUST accept the request, MUST close the Job if the LastDocument Element is asserted, MUST add the supplied Document Data (if any) to the identified Job, and MUST respond to the request.

7.3.1.19 Send<service>Uri

The Send<service>Uri operation allows a client acting for the Job Owner or an Administrator or Operator to input a Digital Document to a Service as part of an already created Job. As such, the Send<service>Uri operation is identical to the Send<service>Document except that a client supplies a URI reference (DocumentUri Element) rather than the Document Data itself. If a Service supports both operations,

clients can use both Send<service>Uri and Send<service>Document operations to add new Documents to an existing multi-Document Job.

As with Send<service>Document, if the Service supports Send<service>Uri but does not support multiple Documents per Job, the Service MUST reject subsequent Send<service>Uri requests associated with the same Job. Similarly, if the Service does support multiple Documents per Job, the Service MUST reject Send<service>Uri requests associated with a given Job after inputs to that Job have been closed. Job inputs can be closed either by a Close<service>Job operation or a Send<service>Document (NOT a Send<service>Uri) request with a 'true' value for the LastDocument Element. Note that the Client may send and the Service must accept a Send<service>Document request with a 'true' value for the LastDocument Element to close input to that Job even if that request includes no Document data.

The Service MUST reject this request and send an appropriate message if:

1. The requestor is not the owner of the identified Job, or is not an Administrator or operator
2. The Service has already closed inputs to the identified Job,
3. The Job is not found
4. The Document size, format and/or compression are not supported by the Service, or
5. The Service does not support the URI Scheme specified.

Otherwise, the Service MUST accept the request, MUST close the Job if the LastDocument Element is asserted, MUST add the Document Data (if any) to the identified Job, and MUST respond to the request. See the Create<system>Job description for discussion of issues relating to excessive delay between multiple Send<service>Uri requests.

7.3.1.20 Set<service>DocumentElements

The Set<service>DocumentElements operation allows a Client, operating for the Job Owner or an Administrator, to set the values of identified Elements of the specified Document within the specified Job. This operation is parallel to the Set<service>JobElements and Set<service>ServiceElements operations and it follows the same rules for validation, but with the target and response Elements relating to a Document rather than a Job or the Service.

The Client must fully identify the Elements to be set as well as the set values. The only settable Elements are those within the Document Ticket. The Document Data is not part of the Document and cannot be changed using this operation. If a Document was originally submitted without a given settable Element that the Set<service>DocumentElements request attempts to set, the Service adds the specified Element to the Document.

If the client identifies a Document Element but does not specify a value for that Element, then the Service MUST remove the Element and all of its values from the Document. The semantic effect of the client supplying the Element with no value in a Set<service>DocumentElements operation MUST be the same as if the Element had not originally been supplied with the Document. This corresponds to the action of the out-of-band value “DeleteElement” in the antecedent IPP Set-Document-Attributes operation. Any subsequent Get<service>DocumentElements or Get<service>Documents request MUST NOT return any Element that has been deleted. However, a client can re-establish such a deleted Document Element with any supported value(s) using a subsequent Set<service>DocumentElements operation.

If the client supplies an Element in a Set<service>DocumentElements request with no value and that Element is not present in the Document object, the Service ignores that supplied Element in the request, does not return the Element in the Unsupported Elements group, and returns the ‘success’ status code, provided that there are no other problems with the request.

The validation of the Set<service>DocumentElements request is performed by the Service as if the Document had been submitted originally with the new Element values (and the deleted Elements

removed); i.e., all modified Document Elements and values must be supported in combination with the Document Elements not modified. If such a Document Creation operation would have been accepted, then the Set<service>DocumentElements MUST be accepted. If such a Document Creation operation would have been rejected, then the Set<service>DocumentElements MUST be rejected and the Document MUST be unchanged. In addition, if any of the supplied Elements are not supported, are not settable, or the values are not supported, the Service MUST reject the entire operation; the Service MUST NOT set just some of the supplied Elements. That is, Set<service>DocumentElements MUST be implemented as an atomic operation; after the operation, all the supplied Elements MUST be set or all of them MUST NOT be set.

The value of JobMandatoryElements supplied in the original Create<service>Job request, if any, MUST have no effect on the behavior of the Set<service>DocumentElements operation. Rather, the Service must consider that any Element or Element value in a Set<service>DocumentElements operation is mandatory. The Service MUST reject any request to set a Document Element to an unsupported value or to a value that would conflict with another Document Element value.

The Service MUST respond to the Set<service>DocumentElements operation as defined for the antecedent Set-Document-Attributes operation in the Standard for IPP Document Objects [PWG5100.5]. Although the Document's current state affects whether the Service accepts or rejects the Set<service>DocumentElements request, the operation MUST NOT change the state of the Document object (since the Document is a passive object and the Document state is a subset of the JobState). For example, if the operation creates a request for unavailable resources, the Job (but not the Document) transitions to a new state.

7.3.1.21 Set<service>JobElements

The Set<Service>JobElements operation allows a Client operating for the Job Owner or an Administrator, to set the values of identified Elements of the specified Job. The Client must fully identify the Elements to be set as well as the set values. In the response, the Service returns success or rejects the entire request with indications of which Element or Elements could not be set to the specified values.

This operation is parallel to the Set<service>DocumentElements and Set<service>ServiceElements operations and it follows the same rules for validation, but with the target and response Elements relating to a Job rather than a Document or the Service

If the client identifies a Job Element but does not specify a value for that Element,, then the Service MUST remove the Element and all of its values from the Job. The semantic effect of the client supplying the Element with no value in a Set<service>JobElements operation MUST be the same as if the Element had not originally been supplied with the Job. This corresponds to the action of the out-of-band value “DeleteElement” in the antecedent IPP Set-Job-Attributes operation. Any subsequent Get<service>JobElements or Get<service>Jobs request MUST NOT return any Element that has been deleted. However, a client can re-establish such a deleted Job Element with any supported value(s) using a subsequent Set<service>JobElements operation.

If the client supplies an Element in a Set<service>JobElements request with the DeleteElement value and that Element is not present on the Job object, the Service ignores that supplied Element in the request, does not return the Element in the Unsupported Elements group, and returns the ‘success’ status code, provided that there are no other problems with the request.

The validation of the Set<service>JobElements request is performed by the Service as if the Job had been submitted originally with the new Element values (and the deleted Elements removed); i.e., all modified Job Elements and values must be supported in combination with the Job Elements not modified. If such a Job Creation operation would have been accepted, then the Set<service>JobElements request MUST be accepted. If such a Creation operation would have been rejected, then the Set<service>JobElements MUST be rejected and the Job MUST be unchanged. In addition, if any of the supplied Elements are not supported, are not settable, or the values are not supported, the Service

MUST reject the entire operation; the Service MUST NOT partially set some of the supplied Elements. In other words, after the operation, all the supplied Elements MUST be set or none of them MUST be set, thus making the Set<service>JobElements an atomic operation.

The value of JobMandatoryElements supplied in the original Create<service>Job request, if any, MUST have no effect on the behavior of the Set<service>JobElements operation. Rather, the Service must consider that any Element or Element value in a Set<service>JobElements operation is mandatory. The Service MUST reject any request to set a Job Element to an unsupported value or to a value that would conflict with another Job Element value.

The Service MUST accept or reject the Set<service>JobElements operation according to the rules defined for the antecedent Set-Job-Attributes operation in Internet Printing Protocol (IPP):Job and Printer Set Operations [RFC3380].

7.3.1.22 SuspendCurrent<service>Job

The SuspendCurrent<service>Job operation allows a Client operating for the Job Owner or an Administrator, to suspend a Job by setting a condition in a Job that is currently in the Processing or ProcessingStopped state. This condition, reflected by the JobSuspended value in that Job's JobStateReasons Element, causes that Job to be in the ProcessingStopped state. The Service is able to processes other Jobs normally, provided that no other inhibiting conditions exist. Note that a Job may be ProcessingStopped state for other reasons and that, once it has been suspended, the Job will remain in the ProcessingStopped state even after the other conditions have been removed.

There is the potential that the current Job may have changed between the time a client requests this operation and the time the Service implements it. Therefore, if the intent is to suspend a particular Job, the Client can include an optional JobId parameter in the request.

The target Job is:

- The Job identified by the JobId, if included in the request
- If the JobId is not included in the request, any Jobs in the Processing or ProcessingStopped state to which the requestor has access rights.

The Service MUST reject the request and send an appropriate message if:

a. There is no target Job in the Processing or ProcessingStopped state to which the requestor has access rights.
b. The target Job or all potential target Jobs have already been suspended.

The Service MUST accept the request, cancel any target Job(s) that have not been previously suspended, and return an appropriate message if:

1. The target JobId is included in the request and that Job is currently in the Processing or ProcessingStopped state (but is not suspended), and the requestor has access rights,
2. If no JobId is included and the requestor has access rights to the Job that is currently in the Processing or ProcessingStopped state (but is not suspended), the Service MUST accept the request and suspend that Job.
3. If more than one Job is in the Processing or ProcessingStopped state (but are not suspended), all such Jobs MUST be suspended unless the operation request included the optional JobId, in which case only the identified target Job MUST be suspended.
4. If the JobId is included in the request and that Job is not currently in the Processing or ProcessingStopped state; or if the JobId is not included and there is no Job currently in the Processing or ProcessingStopped state, the Service MUST reject the request and return the appropriate error code.

5. If the JobId is included in the request and that Job has been suspended; or if no JobId is included and is currently in the Processing or ProcessingStopped state, the Service MUST reject the request and return the appropriate error code.

The Resume<service>Job operation causes a suspended Job to be released. If a Service supports SuspendCurrent<service>Job operation, it MUST support the Resume<service>Job operation, and vice-versa.

7.3.2 Administrative Service Operations

Administrative Service operations directly affect the Service as a whole or affect the Jobs of multiple Job Owners. Access is reserved for Administrators or Operators. The MFD Administrative Service Operations are listed in Table 79 and are described below.

Table 79 Administrative Operations

Operation	Request Parameters (Note 2)	Response Parameters (Note 3)	Note
Cancel<service>Jobs	ElementsNaturalLanguage(optional), JobIds(optional), Message (optional) RequestingUserName	JobIds (optional)	1
Disable<service>Service	ElementsNaturalLanguage(optional) Message (optional), RequestingUserName		
Enable<service>Service	ElementsNaturalLanguage(optional), Message (optional) RequestingUserName	-	
HoldNew<service>Jobs	ElementsNaturalLanguageRequested (optional), JobHoldUntil | JobHoldUntilTime, Message(optional), RequestingUserName		
Pause<service>Service	ElementsNaturalLanguageRequested (optional), Message(optional), RequestingUserName		
Pause<service>ServiceAfterCurrentJob	ElementsNaturalLanguageRequested (optional), Message(optional), RequestingUserName		
Promote<service>Job	ElementsNaturalLanguageRequested (optional), JobId, Message(optional), PredecessorJobID(optional), RequestingUserName		
ReleaseNew<service>Jobs	ElementsNaturalLanguageRequested (optional), Message(optional), RequestingUserName		
Restart<service>Service	ElementsNaturalLanguageRequested (optional), IsAcceptingJobs| IsAcceptingResources (optional), Message(optional), RequestingUserName		
Resume<service>Job	ElementsNaturalLanguageRequested(optional), JobId, Message(optional), RequestingUserName		
Resume<service>Service	ElementsNaturalLanguageRequested(optional), Message(optional), RequestingUserName		
Set<service>ServiceElements	DefaultJob Ticket(optional), RequestingUserName ElementsNaturalLanguageRequested (optional), Capabilities(optional), CapabilitiesReady(optional), Description(optional), Message(optional),	Unsupported Elements(optional)	
Shutdown<service>Service	ElementsNaturalLanguageRequested(optional), Message(optional), RequestingUserName		4

Note 1: Cancel<service>Jobs response includes identified but un-cancellable Jobs

Note 2: The RequestingUserName, is used by the Service to determine whether the requestor is an Administrator, Operator or the Job Owner and is therefore authorized to make the request. Some implementations may require further authentication of the requestor's identity. If the requestor is not determined to have access, the Service MUST reject the request.

Note 3: All responses must correlate to request and indicate whether request was successful or failed.

Note 4: Forcing Service Shutdown may also force the state of any active Jobs to Aborted.

7.3.2.1 Cancel<service>Jobs

The Cancel<service>Jobs operation allows the Operator or Administrator of the Service to cancel all identified non-Terminated Jobs or, if no specific Jobs are identified in the request, to cancel all non-Terminated Jobs in the Service. It differs from the Cancel<service>Job operation in that it works on a number of Jobs at once. If, following the legal Job state Transitions in Table, the Service cannot successfully cancel all explicitly or implicitly requested Jobs that are not already in the terminated state it MUST NOT cancel any Jobs but MUST return an error code. In this case, the Service MUST also return the list of JobIds for those Jobs that were explicitly identified in the request but could not be canceled.

The set of candidate Jobs to be canceled is specified by the supplied JobIds. If no JobIds are supplied, it is implicit that all Jobs that are not in a Terminating state are to be canceled. As with all Administrative operations, the Service MUST check the access rights of the requesting user. Provided that the requester has access rights, the Service MUST check the current state of each of the candidate Jobs. If any of the candidate Jobs cannot be canceled, the Service MUST NOT cancel any Jobs and MUST return the indicated error status code along with the list of offending JobId values. If there are no Jobs that cannot be canceled, the Service MUST transition each identified Job to the indicated new state as defined for the antecedent Cancel-Jobs operation in paragraph 6.1 of Standard for Internet Printing Protocol (IPP):9 Job and Printer Extensions Set 2 [PWG5100.11].

7.3.2.2 Disable<service>Service

The Disable<service>Service operation prevents the Service from creating any new Jobs by negating the IsAcceptingJobs Element. This operation has no effect upon the Service State and the Service is still able to process operations other than Create<service>Job. All previously created or submitted Jobs and all Jobs currently processing continue unaffected.

If the requestor is determined to have proper access, the Service MUST accept this request and MUST negate the IsAcceptingJobs Element.

The IsAcceptingJobs Element value is reaffirmed by the Enable<service>Service operation. If an implementation supports Disable<service>Service it must also support Enable<service>Service and vice-versa.

7.3.2.3 Enable<service>Service

The Enable<service>Service operation asserts the IsAcceptingJobs Element to allow the Service to accept new Create<service>Job requests. The operation has no effect upon the Service State or any other operation requests the Service may receive.

If the requestor is determined to have proper access, the Service MUST accept this request and MUST assert the IsAcceptingJobs Element. The Service MUST then be able to accept and implement Create<Service>Job requests, provided that no other inhibiting condition exists.

If a Service implementation supports the Disable<service>Service operation, then it must also support Enable<service>Service operation and vice-versa.

7.3.2.4 HoldNew<service>Jobs

The HoldNew<service>Jobs operation allows a client to prevent any new Jobs from being eligible for scheduling by forcing all newly-created Jobs to the PendingHeld state with a JobHoldUntil or JobHoldUntilTime Job Processing Element added, depending upon the Element supplied with the HoldNew<service>Jobs operation request. The operation has the same effect as a Hold<service>Jobs operation except that any Jobs in the Pending or Processing state when the HoldNew<service>Jobs

request is accepted are allowed to go to completion, provided that no other conditions or operations prevent this.

The JobHoldUntil parameter allows a client to specify holding new Jobs indefinitely or until a specified named time period. The JobHoldUntilTime parameter allows a client to hold new Jobs until a specified time. Provided that the requestor is authorized and the operation and requested parameters are supported, a Service MUST accept a HoldNew<service>Jobs request and MUST add the supplied 'JobHoldUntil' or JobHoldUntilTime Element to the Jobs. This HoldNew<service>Job condition may be cleared by a ReleaseNew<Service>Jobs operation.

If the HoldNewJobs operation is supported, then the ReleaseNew<Service>Jobs operation MUST be supported, and vice-versa

7.3.2.5 Pause<service>Service

The Pause<service>Service operation allows a client to send the Service to the Stopped state. In this Service state, the Service MUST NOT advance any Job to Job Processing state. Depending on implementation, the Pause<service>Service operation MAY also stop the Service from continuing to process any current Job, sending the Job to the ProcessingStopped state. That is, depending upon implementation, any Job that is currently in the Processing state may be sent to the ProcessingStopped state as soon as the implementation permits; or the Job may continue to a termination state as determined by other conditions. The Service MUST still accept CreateJob operations to create new Jobs, provided that there are no other conditions preventing it.

If the Pause<service>Service operation is supported, then the Resume operation MUST also be supported, and vice-versa.

Service State transitions resulting from a Pause<service>Service operation are the same as defined for the antecedent Pause-Printer operation in paragraph 3.2.7 of IPP/1.1: Model and Semantics [RFC29110. The Pause<service>Service action should be done as soon as the possible after the request is accepted. If the implementation will take more than negligible time to stop processing (perhaps to finish processing the current Job), the Service may remain in the ‘Processing’ state but MUST add the 'MovingToPaused' value to the Service’s StateReasons Element. When the Service transitions to the 'Stopped' state, it removes the 'MovingToPaused' value and adds the 'Paused' value to the Service’s StateReasons Element. If the implementation permits the current Job to stop in mid processing, the Service transitions directly to the ‘Stopped’ state with the Service’s StateReasons Element set to the 'Paused' value and the current Job transitions to the 'ProcessingStopped' state with the JobStateReasons Element set to the 'Stopped' value.

For any Jobs in the 'Pending' or 'PendingHeld' state, the ‘Stopped' value of the Jobs' JobStateReasons Element also applies. However, the Service need not update those Jobs' JobStateReasons Element and need only return the 'Stopped' value when those Jobs are queried (so-called lazy evaluation).

Provided that the requestor is authorized, the Service MUST accept the Pause<service>Service request in any Service state and act as defined for the antecedent Pause-Printer operation in paragraph 3.2.7 of IPP/1.1: Model and Semantics [RFC29110].

7.3.2.6 Pause<service>ServiceAfterCurrentJob

The Pause<service>ServiceAfterCurrentJob operation allows a client to stop the Service from processing any Jobs once any Jobs currently in Processing are completed. This operation has no effect on the current Jobs and the Service MUST complete the processing of the current Jobs, provided that no other condition or operations preclude it. The Service MUST still accept **CreateJob** operations to create new Jobs, but MUST not cause any Jobs to enter 'Processing'. If the Pause<service>ServiceAfterCurrentJob operation is supported, then the Resume<service>Service operation MUST also be supported.

Service State transitions resulting from a Pause<service>ServiceAfterCurrentJob operation are as identified for the antecedent Pause-Printer-After-Current-Job operation in IPP: Job and Printer Operations [RFC3998]. Note that, in implementations where the Service implementation is not able to pause Jobs currently in the Processing state, the response to the Pause<service>ServiceAfterCurrentJob request and the Pause<service>Service request are exactly the same.

If the implementation will take more than negligible time to finish processing the current Jobs, the Service will remain in the Processing state and must add the 'MovingToPaused' value to the Service's StateReasons Element. When the Service transitions to the 'Stopped' state, it removes the 'MovingToPaused' value and adds the 'Paused' value to the Service's StateReasons Element.

For any Jobs in the 'Pending' or 'PendingHeld' state, their state is unchanged but the JobStateReasons Element must be set to the 'Stopped' value. However, the Service need not update those Jobs' JobStateReasons Element and only need return the 'Stopped' value when those Jobs are queried (so-called lazy evaluation).

Provided that the requestor is authorized, the Service MUST accept the request in any Service state and MUST transition the Service to the indicated new State as follows before returning the operation response as defined for the antecedent Pause-Printer-After-Current-Job operation in IPP: Job and Printer Operations [RFC3998].

7.3.2.7 Promote<service>Job

The Promote<service>Job operation schedules the identified Job to be processed next, after the currently processing Jobs or, if the request includes the predecessor JobId, immediately after the identified predecessor Job. The Promote<service>Job operation is a combination of the IPP Promote-Job and Schedule-Job-After operations. If the predecessor Job is not specified, it acts in the same way as the antecedent IPP Promote-Job operation. If the predecessor Job is specified, it acts the same way as the antecedent IPP Schedule-Job-After operation.

The identified target Job must be in the 'Pending' state. If the identified target Job is not in the 'Pending' state or if the predecessor Job is identified and it is not in the 'Pending', 'Processing' or 'ProcessingStopped' state, the Service MUST reject the request and return an appropriate status code. If the Promote<service>Job request is accepted, the target Job MUST be processed immediately after the current Jobs or identified predecessor Job reaches a Termination state (Canceled, Completed or Aborted)

Note that the action of this operation is consistent even if a previous Promote<service>Job Request has caused some other Job to be scheduled after the current or predecessor Job; that is, within the rescheduling time limitations of the Service, the Job identified in the last Promote<service>Job Request accepted will be processed next.

7.3.2.8 ReleaseNew<service>Jobs

The ReleaseNew<service>Jobs operation allows a client to remove the condition initiated by HoldNew<service>Jobs and to release all Jobs previously forced to a PendingHeld state by the HoldNew<service>Jobs initiated condition so that these Jobs are eligible for scheduling. This is done by removing the 'JobHoldUntilSpecified' and 'JobHeldByService' values from the Job's JobStateReasons Element and changing the Jobs' states to 'Pending'.

Provided that the requestor is authorized, the Service MUST accept this request in any Service state and the Service MUST remove the 'JobHoldUntilSpecified' value from the Job's JobStateReasons Element for any Job previously forced to a PendingHeld state by the HoldNew<service>Jobs initiated condition.

If the ReleaseNew<service>Jobs operation is supported, then the HoldNew<service>Jobs operation MUST be supported, and vice-versa.

7.3.2.9 Restart<service>Service

The Restart<service>Service operation causes a Service in any state, even a previously shut down instance of a Service, to be initialized and set to the Idle state, provided that no errors occur or conditions exist that would prevent normal operation. The handling of Jobs that were in the Processing, Pending, PendingHeld, and ProcessingHeld states state prior to Restart is implementation dependent, but a Service Restart MUST be performed as gracefully as possible and in a way preserving the content and integrity of any non-terminated Jobs. Job history data, if supported, SHOULD also be preserved; a particular Service may make this mandatory.

Provided that the requestor is authorized, the Service MUST accept the request Restart<service>Service regardless of its current state. Providing that no conditions exist that would normally prevent these actions, the Service MUST reinitialize its State to Idle, clear the StateReasons Element and set the IsAcceptingJobs Element to true.

7.3.2.10 Resume<service>Service

The Resume<service>Service operation allows a client to cause the Service to resume scheduling Jobs after scheduling has been paused. Provided that the requestor is authorized and the Service supports this operation, a Service MUST accept a Resume<service>Service request regardless of the current Service state, corresponding to the actions defined for the antecedent Resume-Printer operation in Internet Printing Protocol/1.1: Model and Semantics [RFC2911]. If there are no other reasons why the Service is in the Stopped state, this operation returns the Service from the Stopped state to the Idle or Processing state from which it was paused, and removes the 'Paused' value to the Service's StateReasons Element.

If the Resume<service>Service operation is supported, then the Pause<service>Service operation MUST be supported, and vice-versa.

7.3.2.11 Set<service>ServiceElements

The Set<service>ServiceElements operation allows a Client to set the values of identified Elements in the Service, provided that they are settable. Settable Elements may be in Service Capabilities, Service Configuration, Service Description and DefaultJob Ticket but not in Service Status.

The Service MUST reject the entire request with indications of which Element or Elements could not be set if a client request attempts to:

1. Set a non-settable Element (including an Element not in the Service Capabilities, Service Configuration, Service Description or DefaultJob Ticket groups, a read-only Element, and an Element not supported or not supported as a writable Element in the specific Service implementation)
2. Set a settable Element to an invalid value or to a value that conflicts with the values of other Service Elements, including Elements being set in the same request.
3. Set a greater number of Elements in one operation than are supported by the Service implementation (a Service implementation need not support set of more than one Element at a time).

A Set<service>ServiceElements operation that specifies an Element but provides no value for that Element is not an error but rather a request to eliminate that Element and whatever value it has.

If there is no reason to reject setting all of the specified Elements to the specified values or elimination of the Element, the Service MUST accept this operation request when it is in the Idle or Stopped state, and SHOULD accept the request when it is in the Processing state.

If the Service accepts the request, only those Elements specified in the request are changed unless the definition of one or more of the set Elements explicitly specifies an effect upon some other Element.

7.3.2.12 Shutdown<service>Service

The Shutdown<service>Service operation forces the Service to the 'Down' state from any state that it is in, in an orderly manner. That is, the Service MUST stop accepting any further client requests, and MUST stop scheduling Jobs for processing as soon as the implementation allows, although it SHOULD complete the processing of any currently processing Jobs. Once down, the Service will no longer respond to any Client requests other than Restart<service>Service request. As with the antecedent IPP Shutdown-Printer operation all Jobs MUST be preserved. As with Restart<service>Service, Service shutdown must be performed as gracefully as possible and in a way in preserving the content and integrity of any non-terminated Jobs. Job history data, if supported, SHOULD also be preserved.

Once shut down, a Service can be roused from its Down state by a Restart<service>Service operation. If a Service implementation supports Shutdown<service>Service it must also support Restart<service>Service and vice-versa. In the down state, the only operation request that a service will respond to is a Restart<service>Service operation.

Provided that the requestor is authorized, the Service MUST accept this operation and following an orderly progression, transition to the Down state regardless of the current state of the Service.

8 Counters & Timers

Counters and timers provide necessary information relating to cost accounting, component wear, utilization efficiency, reliability and other factors. Some of the items being counted or periods being timed are specific to a particular Service or Subunit and are discussed in conjunction with that Service or Subunit. However, because MFDs are ultimately concerned with Hardcopy Documents and involve typical computing and communication components, many of the counter and timer types are common to several layers of Elements in the MFD Model as well as in the System. Counters and timers of a particular parameter may be with respect to the MFD System, a Service, or Subunits. Many counters and timers may also apply to a Job or to a Document. The potential for a counter or timer to be present is indicated in the models of the System, Service, Subunit, Job and Document.

These counter types are defined in the PWG Standardized Imaging System Counters specification [PWG5106.1], and are briefly identified in this Chapter. Functionally, counters are classified according to four main areas, as shown below, with the major Work group being further divided.

Work Counters: This category measures work that is produced or processed by the imaging Service or System as its primary function. The Work counters are subdivided into six groups:

a. **Work Totals**: The sum total of counts in the remaining five Work group counters.
b. **Datastream:** Counters associated with work performed directly in processing datastream content. That is, these counters reflect the desired primary output of the Service or System.
c. **Auxiliary**: Counters associated with auxiliary content (e.g., banner sheets, confirmations, and separator sheets) or units of work generated internally by the system or service (e.g., reports, start-up, calibration).
d. **Waste Counters**: Counters associated with non-productive work or waste generated by the Imaging System
e. **Maintenance** Counters: Counters associated with all work performed and waste generated while the system is in maintenance mode.
f. **Other**: Counts accumulated in a WorkTotals counters that are not otherwise discriminated as Datastream, Auxiliary, Waste or Maintenance.

Media Used Counters: Measure of the sheets of defined media types used by an imaging service or consumed across multiple services during the imaging process.
Monitoring Counters: Measure of raw traffic and record of error and fault information associated with a service used to determine workload and operating conditions at a high level.
Availability Counters: Measure of the number of times a service is in a particular state. Availability counters are used to measure availability of a system or service.

The counter Elements associated with the System Status are represented in Figure 115. Although not evident from this diagram, the WorkTotals counter reflects the totals from only the “work” counters: DataStream, Auxiliary, Maintenance and Waste (and possibly some vendor specific “work” counters) Media Used, Monitoring and Availability counters (and possibly some vendor specific other counters.)are independent. This same ‘counters’ structure is used for System, Services, Subunits, Jobs and Documents.

This chapter describes this structure and that of each of the counter types in the structure.

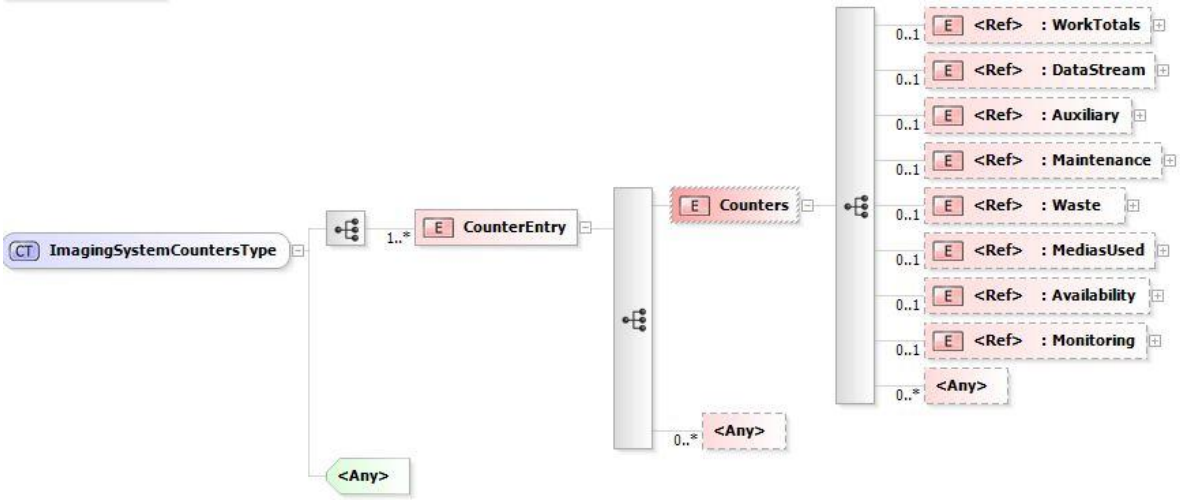

Figure 115 System Counters

The counter structure can exist at different levels in the model and there should be a relationship between counters. For example, the summation of the Document Counters for all Documents in a Job should equal the values of the corresponding Job Counters. The summation of all Job Counters should equal the values of the corresponding Service Counters. The counters for the System are the accumulation of all counts of that type from all of the Services in the System. The WorkTotals values are a different summation, namely that of all of the Work Counters at the same level. These relationships are represented inFigure 116.

The Counters in the MFD Model are all “lifetime” counters; that is, they are not settable or resettable but accumulate counts for the lifetime of the Document, Job, Service or System to which they apply.

8.1 Work Counters (WorkTotals, DataStream, Auxiliary, Waste and Maintenance)

The Work counters reflect an Imaging System, Service or Subunit in performing its primary function. For example, the Work done by a Service producing hardcopy output is measured primarily in terms of Impressions; the Work done by a Service producing Digital Documents is measured in terms of Images. Facsimile Services work is characterized by Messages. Because most Services deal with Digital Data communication, most include a count of Input and/or Output KOctets.

Note that:

- An Impression is the content imposed upon a one side of a Media Sheet by a marking engine, independent of the number of times that the sheet side passes any marker.
- A Blank Impression results when no content is imposed on a Media Sheet side, although the sheet side has passed though the Marker in a way that could have imposed content. The reverse side of a Media Sheet which has been printed in a simplex [one sided] mode is not considered to have a blank impression.
- An Image is the digital representation of a virtual media sheet side.
- A Message is a single application protocol request or response (that may consist of multiple application protocol data units) received or sent by Service such as EmailIn or FaxOut.
- KOctets is the unit of measurement of the amount of data (in 1024 octets) that was consumed (input) or produced (output) by the Service or System.

Each of the Work Counters (WorkTotals, DataStream, Auxiliary, Waste and Maintenance), at each level where counters may exist, has the same structure, shown in Figure 117, with constituent Elements listed in Table 80. Note however that all counter types are optional.

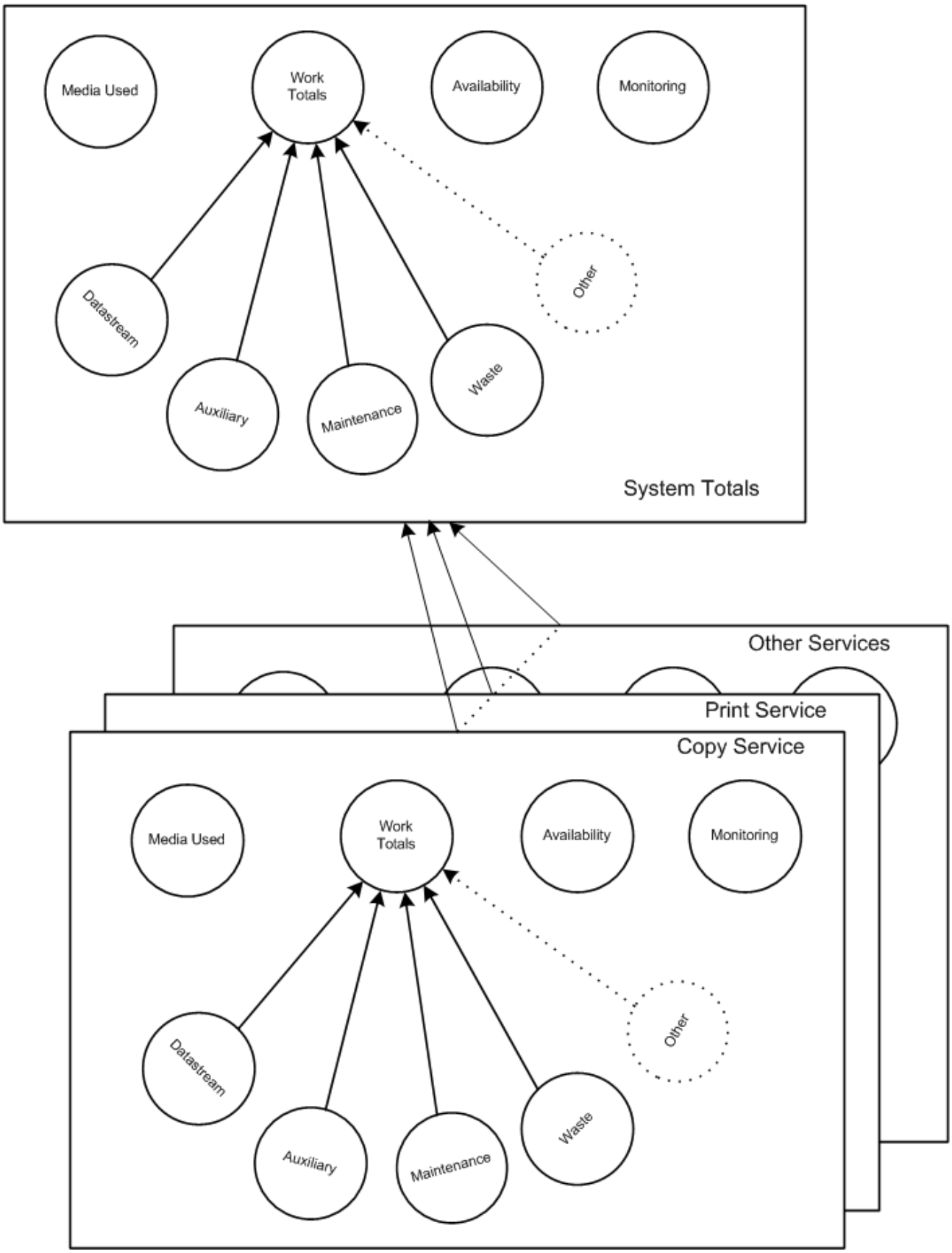

Figure 116 Counters, Showing the Derivation of Service Totals and System Totals

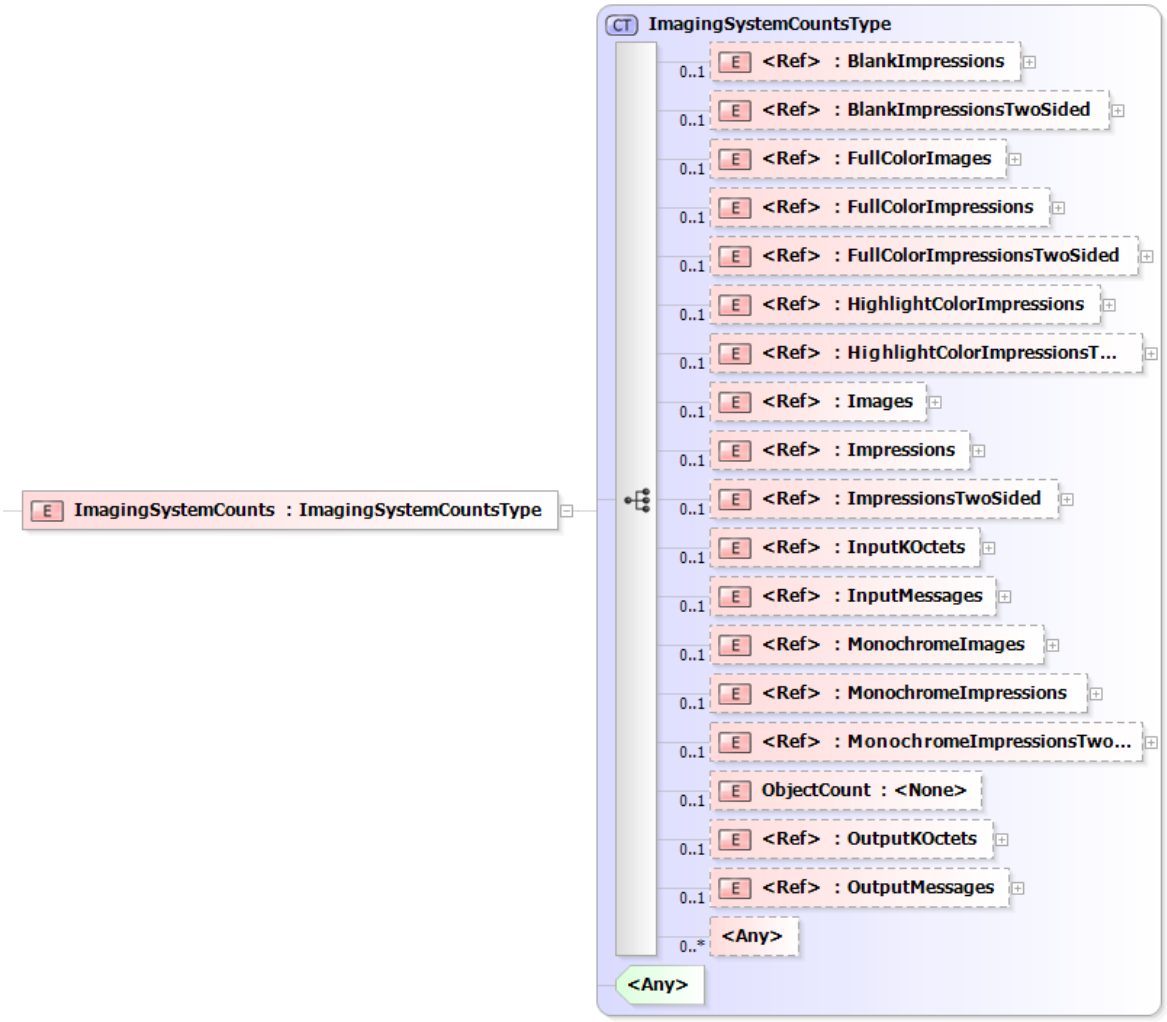

Figure 117 Work Counter Counts Type

Table 80 Counter Elements Used in the Various Service and System Work Counters

Element	DataType	Description *
BlankImpressions	int	Total of blank impressions, simplex and duplex
BlankImpressionsTwoSided	int	Number of instances where both sides of a sheet printed in duplex are blank
FullColorImages	int	Number of Images requiring three or more pixel colors to represent
FullColorImpressions	int	Number of Impression requiring three or more colorants
FullColorImpressionsTwoSided	int	Number of instances where both sides of a sheet printed in duplex are FullColorImpressions
HighlightColorImpressions	int	Number of Impression requiring a black colorant plus one other colorant
HighlightColorImpressionsTwoSided	int	Number of instances where both sides of a sheet printed in duplex are HighlightColorImpressions
Images	int	Total images of all types
Impressions	Int	Total Impressions of all types
ImpressionsTwoSided	int	Total number of instances of all types where both sides of a sheet printed in duplex
InputKOctets	int	Quantity of incoming data In units of 1024 octets
InputMessages	int	Total number of messages received by the service

Element	DataType	Description *
MonochromeImages	int	Number of Images requiring one pixel color to represent
MonochromeImpressions	int	Number of Impression requiring one colorant
MonochromeImpressionsTwoSided	int	Number of instances where both sides of a sheet printed in duplex are MonochromeImpressions
OutputKOctets	int	Quantity of outgoing data In units of 1024 octets
OutputMessages	int	Total number of messages sent by the service
Any	various	Extension point for Counts

*These Elements are defined in PWG 5106.1-2007, "The Printer Working Group (PWG) Standardized Imaging System Counters 1.1" [PWG5106.1]

8.2 MediaUsed Counters

The Media Used counters track the number of sheets of defined media types used by an Imaging System, Service, Job or Document. These counts are necessary for billing and accounting, and for stock control. Media Counter structure is shown in Figure 118 with constituent Elements described in Table 81 Media Used Counter Elements.

Figure 118 Media Used Counters

Table 81 Media Used Counter Elements

Element	Datatype	Description*
MediaUsedAccountingKey	keyword	non-localizable keyword ensuring machine readable, locally unique identification of the specific media. This Element MUST clearly distinguish different instances of the same media size (for example, by including specific media color, weight, etc.)
MediaUsedBlankSheets	int	Count of instances this media had blank impression
MediaUsedFullColorSheets	int	Count of instances this media had full color impression
MediaUsedHighlightColorSheets	int	Count of instances this media had highlight color impression
MediaUsedMediaInfo	string	Human readable media description
MediaUsedMonochromeSheets	int	Count of instances this media had monochrome impression
MediaUsedSizeName	keyword	See PWG Media Standardized Names [PWG 5101.1)
MediaUsedTotalSheets	int	Count of all instancs that this media was used called MediaUsedSheets in Counter Spec.[PWG5106.1]
Any	various	Extension point for MediaUsed

*These Elements are defined in PWG 5106.1-2007, "The Printer Working Group (PWG) Standardized Imaging System Counters 1.1" [PWG5106.1]

8.3 Availability Counters

The Availability counters indicate the availability a System, Service or Subunit by measuring the Down Time, Processing Time, Time In, Maintenance Mode and Total Time. Idle Time can be derived by subtracting the Down, Processing and Maintenance time values from Total Time.

The structure of Availability Counters is represented in Figure 119. The description of the constituent Elements is in Table 82.

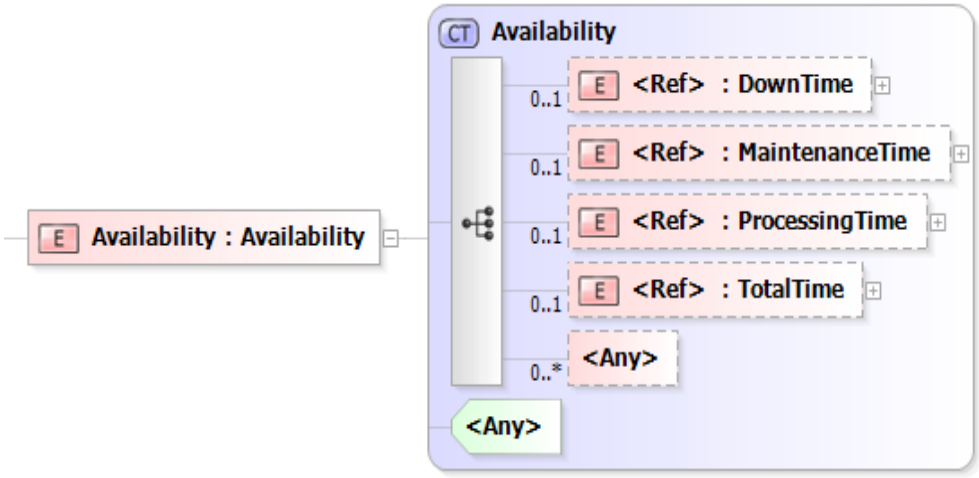

Figure 119 Availability Counters

Table 82 Availability Counter Elements

Element	Datatype	Description
DownTime	int	See PWG 5106.1-2007, "The Printer Working Group (PWG) Standardized Imaging System Counters 1.1" for definitions. Units are in seconds. [PWG5106.1]
MaintenanceTime		
ProcessingTime		
TotalTime		
Any	various	Extension point for Availability

8.4 Monitoring Counters

Monitoring counters give a high level view of the imaging System, Service, and Subunit workload and conditions by recording data flow and warnings and errors conditions. The Monitoring counter structure is represented in Figure 120 with the counter types listed in Table 83 .

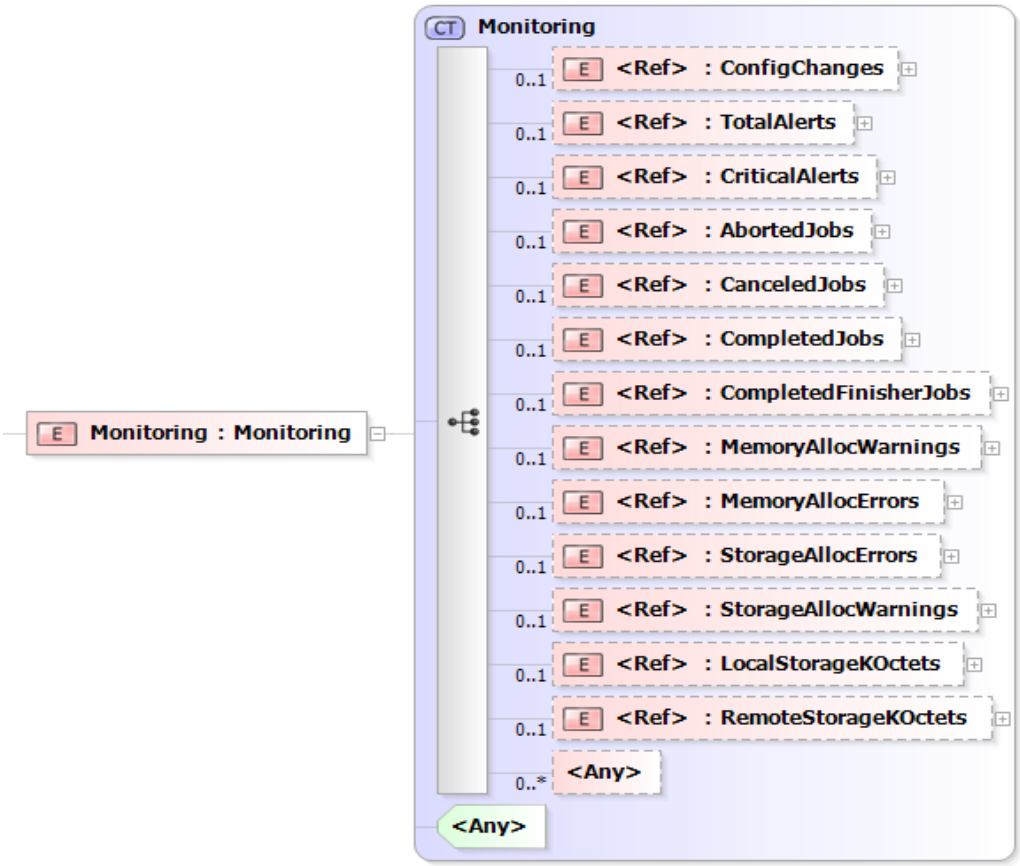

Figure 120 Monitoring Counts

Table 83 Monitoring Count Elements

Element	Datatype	Description
AbortedJobs	int	See PWG 5106.1-2007, "The Printer Working Group (PWG) Standardized Imaging System Counters 1.1" [PWG5106.1]
CanceledJobs	int	
CompletedFinisherJobs	int	
CompletedJobs	int	
ConfigChanges	int	
CriticalAlerts	int	
LocalStorageKOctets	int	
MemoryAllocErrors	int	
MemoryAllocWarnings	int	
RemoteStorageKOctets	int	
StorageAllocErrors	int	
StorageAllocWarnings	int	
TotalAlerts	int	
Any	various	Extension point for Monitoring Counts

9 Conformance

This Document does not define the model of any particular Imaging Service but rather defines Elements, terms, concepts and operations common to several Imaging Service Models. Conformance requirements applicable to each Service Model are defined in the specification for that Service Model.

- MFD Service Model and MFD System Model specifications MUST import the definitions of Elements, terms and semantics from this specification.
- Although when necessary, those MFD Service Model and the MFD System Model specifications may define new Elements, terms or concepts, they MUST NOT attempt to redefine, alter or even restate any of the Elements, terms or semantics defined in this specification.

10 PWG Registration Considerations

This specification and the associated schema will require an update to register extensions to the MFD Service model. Vendors may use extensions in their own namespace until such time as an update to the specification is under review. At that time the extension can be registered with the PWG and included in the PWG standard.

11 Internationalization Considerations

All Element values defined by enumeration (e.g., State) represent keywords. Keywords are never localized by the device. The client may convert the values into a form acceptable to the client. This includes not only localization but also transformations into graphical representation. The Elements with an extensible list of keyword are represented by the union of an enumeration of keywords and a pattern for new values.

Some of the Elements have values that are Service-generated strings (e.g., State Messages). In each operation request, the client identifies a natural language that affects the Service generated strings returned by the Service in operation responses. The Service MUST provide the localized value as requested by the user for any supported natural languages. A request for a language not supported results in a response with the string in the default localization.

The final category of string values are those supplied by administrator or End User (e.g., JobName). No localization is performed on these strings and they are returned in operation responses as set by the administrator or End User.

12 Security Considerations

The exact security model for MFD's and implementation are out-of-scope for this specification. The IEEE 2600™-2008 standard [IEEE2600] defines security requirements for manufacturers, administrators, and others in the selection, installation, configuration, and usage of hardcopy devices, including MFDs. The common security considerations for all MFD Imaging Services outlined in the MFD Service Model Requirements [MFD-REQ] are intended to support the IEEE 2600™-2008 standard. Implementations must preserve the MFD Service Model semantics when addressing security considerations and requirements.

Security considerations specific to each MFD Service are discussed in the specification for the corresponding Service Model.

13 References

[CIM-PRINTER]
DMTF CIM Schema Printer class CIM_Printer.mof in any version of [CIM-SCHEMA]

[CIM-SCHEMA]
DMTF CIM Schema Repository http://dmtf.org/standards/cim

[DSP0004]
DMTF Common Information Model (CIM) Infrastructure, DSP0004, March 2010, http://dmtf.org/sites/default/files/standards/documents/DSP0004_2.6.0_0.pdf

[IANA-FIN]
IANA Finisher MIB (originally published in RFC 3806), June 2004, R. Bergman, H. Lewis, I. McDonald, n http://www.iana.org/assignments/ianafinisher-mib

[IANA-PRT]
IANA Printer MIB (originally published in RFC 3805), June 2004, R. Bergman, H. Lewis, I. McDonald, http://www.iana.org/assignments/ianaprinter-mib

[IEEE2600]

IEEE 2600™-2008 IEEE Standard for Information Technology: Hardcopy Device and System Security, 2008

[MFD-REQ]

PWG Multifunction Device Service Model Requirements, September, 2010, N. Chen, I. McDonald, W. Wagner, P. Zehler, ftp://ftp.pwg.org/pub/pwg/informational/req-mfdreq10-20100901.pdf

[MFD_SCHEMA]

PWG Semantic Model Schema v1.127, P. Zehler, I. McDonald, Namespace:http://www.pwg.org/schemas/2011/3/sm/ ftp://ftp.pwg.org/pub/pwg/mfd/schemas/PWG_SM2_ver_1.127.zip

[PWG5100.1]

PWG 5100.1-2001, "Internet Printing Protocol (IPP): “finishings” attribute values extension ", February 5, 2001, T. Hastings, D. Fullman, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippfinishings10-20010205-5100.1.pdf

[PWG5100.2]

PWG 5100.2-2001, "Internet Printing Protocol (IPP): “output-bin” attribute extension ", February 7, 2001, T. Hastings, R. Bergman, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippoutputbin10-20010207-5100.2.pdf

[PWG5100.3]

PWG 5100.3-2001, "Internet Printing Protocol (IPP): Production Printing Attributes Set1", February 12, 2001, K. Ocke, T. Hastings, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippprodprint10-20010212-5100.3.pdf

[PWG5100.4]

PWG 5100.4-2001, "Internet Printing Protocol (IPP): Override Attributes for Documents and Pages ", February 7, 2001, K. Ocke, R. Herriot, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippoverride10-20010207-5100.4.pdf

[PWG5100.5]

PWG 5100.5-2003, "Internet Printing Protocol (IPP): Document Object", October 31, 2003, D. Carney, T. Hastings, and P. Zehler, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippdocobject10-20031031-5100.5.pdf

[PWG5100.6]

PWG 5100.6-2003, Standard for The Internet Printing Protocol (IPP): Page Overrides", October 31, 2003, P. Zehler, R. Herriot, K. Ocke, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ipppageoverride10-20031031-5100.6.pdf

[PWG5100.7]

PWG 5100.7-2003, Standard for The Internet Printing Protocol (IPP): Job Extensions", October 31, 2003, T. Hastings, P. Zehler, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippjobext10-20031031-5100.7.pdf

[PWG5100.10]

PWG 5100.10-2009, "The Internet Printing Protocol Version 2.0", 31 July 2009, R. Bergman, H. Lewis, I. McDonald, M. Sweet, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ipp20-20090731-5100.10.pdf

[PWG5100.11]

PWG 5100.11-2010 "IPP Job and Printer Extensions Set 2 (JPS2)", 30 October 2010, T. Hastings, D. Fullman, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippjobprinterext10-20101030-5100.11.pdf

[PWG5101.1]

PWG 5101.1-2002 Media Standardized Names, February 26, 2002, R. Bergman, T. Hastings, ftp://ftp.pwg.org/pub/pwg/candidates/cs-pwgmsn10-20020226-5101.1.pdf

[PWG5101.2]

PWG 5101.2-2004, "The Printer Working Group (PWG) RepertoireSupported Element", February 1, 2004, H. Lewis, et al, ftp://ftp.pwg.org/pub/pwg/candidates/cs-crrepsup10-20040201-5101.2.pdf

[PWG5105.1]

PWG 5101.2-2004, " Printer Working Group (PWG) Semantic Model ", February 1, 2004, P.Zehler, T. Hastings, S. Albright, ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm10-20040120-5105.1.pdf

[PWG5106.1]

PWG 5106.1-2007, "The Printer Working Group (PWG) Standardized Imaging System Counters 1.1", April 27, 2007, H. Lewis, I. McDonald, J. Thrasher, W. Wagner, and P. Zehler, ftp://ftp.pwg.org/pub/pwg/candidates/cs-wimscount11-20070427-5106.1.pdf

[PWG5106.2]

PWG 5106.2-2006 "Web-based Imaging Management Service V1.0/Abstract Protocol, April 21, 2006. H. Lewis, I. McDonald and W. Wagner, ftp://ftp.pwg.org/pub/pwg/candidates/cs-wims10-20060421-5106.2.pdf

[PWG 5106.4]

PWG 5106.4-2011 "Power Management Model for Imaging Systems 1.0", February 14, 2011, I. McDonald, ftp://ftp.pwg.org/pub/pwg/candidates/cs-wimspower10-20110214-5106.4.pdf

[PWG5108.2]

PWG 5108.2-2009, Network Scan Service Semantic Model and Service Interface v1.0", 10 April 2009, N.Chen, P. Zehler, ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm20-scan10-20090410-5108.02.pdf

[RFC 1696]

RFC 1696, “Modem Management Information Base (MIB) using SMIv2”, August 1994, J. Barnes, L. Brown, R. Royston, S. Waldbusser, ftp://ftp.RFC-editor.org/in-notes/RFC1896.txt

[RFC2119]

RFC 2119, “Key words for use in RFCs to Indicate Requirement Levels”, March 1997, S. Bradner, ftp://ftp.RFC-editor.org/in-notes/RFC2119.txt

[RFC2396]

RFC 2396, “Uniform Resource Identifiers (URI): Generic Syntax”, August 1998, T. Berners-Lee, R. Fielding, L. Masinter, ftp://ftp.RFC-editor.org/in-notes/RFC2396.txt

[RFC2426]

RFC 2426, “vCard MIME Directory Profile”, September 1998, F. Dawson, T. Howes, www.ietf.org/RFC/RFC2426.txt

[RFC2617]

RFC 2617, “HTTP Authentication: Basic and Digest Access Authentication“June 1999, ,J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, http://www.ietf.org/RFC/RFC2617.txt

[RFC 2790]

RFC 2790, “Host Resources MIB, March 2000, S. Waldbusser, P. Grillo, http://www.ietf.org/RFC/RFC3805.txt

[RFC 2863]

RFC 2863, “The Interfaces Group MIB”, June 2000, K. McCloghrie, F. Kastenholz, http://www.ietf.org/RFC/RFC2863.txt

[RFC2911]

RFC 2911 “Internet Printing Protocol/1.1 Model and Semantics”, September 2000, T. Hastings, R. Herriot, R. deBry, S. Isaacson, P. Powell, ftp://ftp.RFC-editor.org/in-notes/RFC2911.txt

[RFC3066]

RFC 3066 “Tags for the Identification of Languages”, January 2001, H. Alvestrand, http://www.ietf.org/RFC/RFC3066.txt

[RFC3192]

RFC 3192 “Minimal FAX address format in Internet Mail”, October, 2001, C. Allocchio. http://www.ietf.org/RFC/RFC3192.txt

[RFC3380]

RFC 3380 “Internet Printing Protocol (IPP): Job and Printer Set Operations”, September 2002, C. Kugler, H. Lewis, T. Hastings, R. Herriot, http://www.ietf.org/RFC/RFC3380.txt

[RFC 3805]

RFC 3805, “Printer MIB v2”, June 2004, R. Bergman, H. Lewis, I. McDonald, http://www.ietf.org/RFC/RFC3805.txt

[RFC 3806]

RFC 3806, “Printer Finishing MIB”, June 2004, R. Bergman, H. Lewis, I. McDonald, http://www.ietf.org/RFC/RFC3806.txt

[RFC3986]

RFC3986, “Uniform Resource Identifier (URI): Generic Syntax”, January 2005, T. Berners-Lee, R. Fielding, L. Masinter, http://www.ietf.org/RFC/RFC3986.txt

[RFC3998]

RFC 3998 “Internet Printing Protocol (IPP): Job and Printer Administrative Operations”, March 2005, C. Kugler, H. Lewis, T. Hastings, http://www.ietf.org/RFC/RFC3998.txt

[RFC4122]

RFC 4122 “A Universally Unique IDentifier (UUID) URN Namespace”, July 2005, P. Leach, M. Mealling, R. Salz, http://www.ietf.org/RFC/RFC4122.txt

[RFC4355]

RFC4355, “IANA Registration for Enumservices email, fax, mms, ems, and sms”, January 2006, R. Brandner, L. Conroy, R. Stastny, http://www.ietf.org/RFC/RFC4355.txt

[RFC4556]

RFC 4556, “Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)” June 2006, L. Zhu, B. Tung, http://www.ietf.org/RFC/RFC4556.txt

[RFC4648]

RFC 4648, “The Base16, Base32, and Base64 Data Encodings”, October 2006, S. Josefsson, http://www.ietf.org/RFC/RFC4648.txt

[RFC5870]

RFC 5870, “A Uniform Resource Identifier for Geographic Locations ('geo' URI)”, June 2010, A. Mayrhofer, C. Spanring, http://www.ietf.org/RFC/RFC5870.txt

[W3C-XSD1]

XML Schema Part 1: Structures Second Edition W3C Recommendation, 28 October 2004, H. Thompson, D. Beech, M. Maloney, N. Mendelsohn, http://www.w3.org/TR/xmlschema-1/

[W3C-XSD2]

XML Schema Part 2: Datatypes Second Edition, 28 October 2004, P. Biron, A. Malhotra, http://www.w3.org/TR/xmlschema-2/

[WS-Scan]

“Scan Service DefinitionVersion 1.0 For Web Services on Devices”, November 2006, Microsoft, M.Fenelon, http://www.microsoft.com/whdc/connect/rally/wsdspecs.mspx

14 Author's Address

William A Wagner
Technical Interface Consulting
Email: wamwagner@comcast.net
Voice: (978)251-4565

Peter Zehler
Xerox Research Center Webster
Email: Peter.Zehler@Xerox.com
Voice: (585) 265-8755
Fax: (585) 265-7441
US Mail: Peter Zehler
Xerox Corp.
800 Phillips Rd.
M/S 128-25E
Webster NY, 14580-9701

Additional contributors:

Nancy Chen, OkiData
Lee Farrell
Ira McDonald, High North
And other members of the PWG MFD Working Group

