`

Project Number	IST-2006-033789
Project Title	Planets
Title of Deliverable	A model for a generic description of preservation action tools and their context
Deliverable Number	D1. Iteration 2, v1.0
Contributing Sub-project and Work-package	SP/PA (PA/2)
Deliverable Dissemination Level	External Public
Deliverable Nature	Report
Contractual Delivery Date	31st August 2007
Actual Delivery Date	31st August 2007
Author(s)	KB-NL

Abstract

This document contains an XML-based language to describe the properties of preservation action tools and their context of use. Within the Planets project this language can serve as a means of communication between a preservation planning system and a tools registry. The schema definition can also serve as a generic model for the description of the functions, expected performance outcomes, and appropriate uses of preservation action tools.

Contributors

Person	Role	Partner	Contribution
Maarten Tacoma		KB-NL	Initiation drafts, processing of comments
Caroline van Wijk		KB-NL	Initiation drafts, processing of comments
Frank Houtman		KB-NL	Initiation drafts, processing of comments

EXECUTIVE SUMMARY

This document contains a draft version of an XML-based language to describe the properties of preservation action tools.

The language, called PTDL (an acronym for Preservation Tool Description Language), will be a format for the exchange of messages between the PA tools registry that is to be developed within the PA/3 work package, and the preservation planning tools provided by the PP sub-project.

It can be used for the following purposes:

- Providing machine-readable information on the contents of the PA tools registry
- Providing human-readable information on tools in the PA tools registry

Development of the language is dependent on several parts of the Planets system, thus on other work packages within the Planets project. At the time this document was written, many of these system parts had not yet been developed. Therefore, the development of the language will be an iterative process during the Planets project and changes to the language presented here will be implemented when information on these specific parts of the system becomes available.

In the first iteration the primary goal was to make an inventory of relevant properties needed in the description of preservation action tools. In this iteration, this initial model has been extended and used to describe tools that were tested by the PA/4 working group. This, as well as the model for the pa tools registry as proposed by the PA/3 workpackage have led to some modifications in the original model (a summary of changes since the last edition can be found in paragraph 2.2). Currently, the applicability of the model is validated more thoroughly by describing various tools (including both tools for objects and tools for environments) and checking whether all the information that needs to be available fits into this model or further modifications will have to be made.

Changes since last edition

In the table below changes between the previous version and the language presented in this document are listed.

Modification	Description	Source	Date
First tests started (describing actual tools used within the project)	The tools that are used in the prototypes for conversion services that are made by the PA/4 workpackage are being used as a test case in order to validate the applicability of the set of properties that have currently been defined. Chapter 6 includes a summary of results of these tests.	-	2007-07-03
'Creator' entity changed to 'Agent'-entity.	In the previous version of the language, the 'Creator' entity was used only to hold information on the creator of a tool. It's used in more places now (supplier of information, author of a piece of documentation) and has been renamed and extended to follow the model for the registry that's being developed in the PA/3 workpackage.	PA/3-D1	2007-06-19
'Documentation' entity added	In the previous version one could only specify whether or not a software product has documentation by means of a URL. In this version the 'documentation' entity has been added to allow more information on a product's documentation to be stored.	PA/3-D1	2007-06-19
'Pathways' have been introduced	In the first version the language was only suitable for describing 'separate' tools, not preservation actions (tools put to a specific purpose, or actions that make use of multiple tools). In this version the 'pathway'-element has been introduced to allow for descriptions of tools and of specific applications of tools.	PA/3-D1	2007-06-19
'Provenance'-entity added	In order to be able to include information on the supplier of the information	PA/3-D1	2007-06-19

	in the registry, a 'provenance'-entity has been added. This can be used to record the person or organisation that added information to or changed information in the registry, as well as the time this happened.		
Compound type for describing input and output formats added	In the first version of this language the file formats a software product can read or write were stored as single PUIDs. To allow for the description of many-to-many conversions, a 'FileSet-entity was introduced.	Feedback on draft version of document by members of PP/4 workpackage	2007-06-21
Number of evaluations has been restricted to one per pathway.	In the previous version it was possible to associate multiple evaluations with a single tool. Since then, it has been decided that evaluations are associated with pathways, not with tools and that only a single evaluation (possibly based on multiple evaluations) can be included for every pathway.	Discussions in PA/2 workpackage	2007-06-19
'Price', 'Compression for download', and Boolean field for 'Supported' removed	After discussion with other project members some properties for tools were considered irrelevant or too dependant on external factors to be maintained. An example is the property 'price' for (commercial) software products. While this can be useful information it's impossible to maintain this information in a tools registry due to changes of prices over time, currencies, etcetera.	Feedback on draft version of document by members of PA/2, PP/4	2007-02-27
'Operation'-cluster of properties was renamed to 'access', and has been changed after discussion with PP/4 members.	The cluster of properties that is used to store information on the invocation of a tool was renamed and slightly changed. This still has to be enhanced in future versions, as more information on the protocols used in the invocation of various tools comes available.	Feedback on draft version of document by members of PP/4 workpackage	2007-02-27
Target and host environments now use the same set of properties	Technical environments that are required by software products in order	Discussions within PA/2 workpackage and	2007-07-30

	to run properly (host environments) and technical environments that are emulated by tools for environments (target environments) are described by the same set of properties now. It is currently an open topic whether or not (an adaptation of) this set of properties will be used to describe technical environments, or whether they will simply be referred to by using a single unique identifier.	feedback from PA/5 workpackage	
Elements have been added to make the language more extendible in the future	The schema now allows for extra elements to be added at a later stage. This was necessary since new child elements will be added at a later stage (for instance, child elements for the 'evaluation' or 'technical environment' entities).	PA/3-D1	2007-06-19
Multiple names allowed for software product	Since software products can be referred to by using different names, and the data model for the tools registry allows multiple names to be associated with a software product, the restriction that software products can have only one name has been removed in this version of the language.	PA/3-D1	2007-06-21
'License'-entity from previous version was renamed and extended to follow PA/3 registry design	The 'license'-entity, used to store information on the intellectual property rights of software products, was renamed to ´Intellectual Property Rights´ and has been extended to follow the data model for the tools registry developed in the PA/3 workpackage	PA/3-D1	2007-07-03

TABLE OF CONTENTS

1. Document overview

1.1 Purpose of this Document

This document consists of a generic description model for preservation action tools and their context of use, and a corresponding XML-based language[1]. It defines the elements used in the language, and introduces the schema definition.

In this chapter, an outline of the document itself is presented and the context of this work within the Planets project is explained. Chapter 2 provides more information on the methodology used in designing the language, while the language itself is presented in chapters 3 and 4. Chapter 5 contains examples of instances of PTDL descriptions.

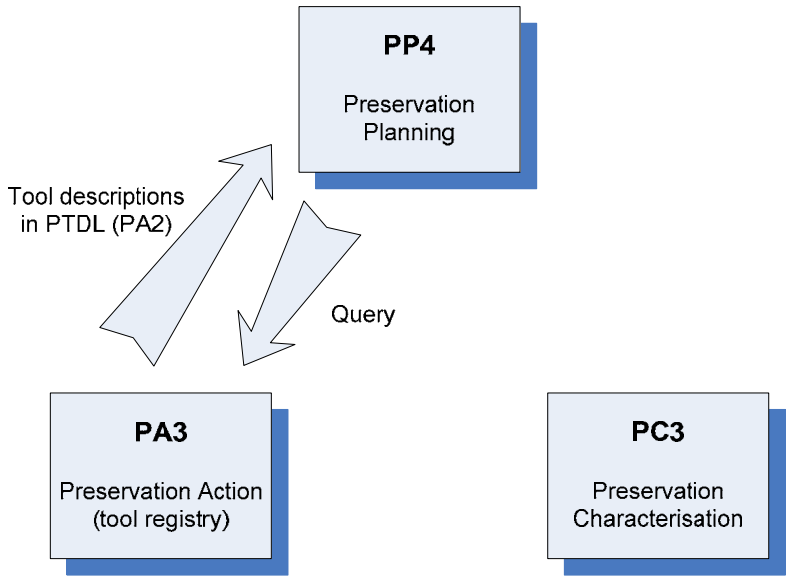
1.2 Scope of this Document

In this document an XML-based language for the description of preservation action tools is presented. These tools include tools for objects and tools for environments, but not characterisation tools as the description of characterisation tools is covered by the PC workpackage. The language will provide a model for describing the functionality, performance outcomes and appropriate uses of preservation action tools.

In the development of the current version, the main goal has been to make an inventory of relevant properties needed to describe PA tools. The properties identified so far have been incorporated in a schema, written in W3C XML Schema[2]. In the current version, the language is not based on the RDF or OWL framework[3], [4]. Also, all data types used in the language (except for the basic ones that are part of the XML-Schema definition) are defined in the schema itself. In subsequent versions, the shared use of schemas for data types within Planets might be desired.

1.3 Context

Within the Planets project, the services developed by the Preservation Planning sub-project need access to information on the tools that are available in the Preservation Action tools registry. Based on this information, the Planning system can take decisions and automatically execute preservation plans.



To achieve this goal, an inventory of significant properties of preservation action tools has been made and an XML-based language was constructed that can be used to present this information in a machine-interpretable way. The language can be used in the exchange of information about stand-alone tools and about 'pathways' (sets of tools used to perform a specific action) and is also designed to store tool information that is intended to be interpreted by humans rather than by machines.

2 Methodology

2.1 Introduction

As part of the Planets project, we aim to provide a mechanism for describing preservation action tools and their context. By defining a set of properties that can be used to describe tools and their context of use, an XML-based markup language is created in which the functions, expected performance outcomes and appropriate uses of preservation action tools can be described.

This chapter describes the methodology used in designing the language and possible overlap with or dependencies on other workpackages. A summary of changes since the last edition is given in paragraph 2.2. Background information on the technical decisions made can be found in chapter 3.

2.2 Summary of changes since last edition

Since the first version of this language was constructed, developments in a number of other workpackages have led to changes in it's design. Several elements have been renamed for consistency with the data model for the tools registry that's being developed within the PA/3 workpackage. This, together with feedback from members of the PP/4 workpackage, has also led to some changes in the model itself. Most notably, the first version of this language didn't include the concept of pathways to be able to describe preservation action tools as used for a particular purpose. In the current version, the language can be used to describe the capabilities of preservation tools in general, but also to describe the capabilities of a tool as it is used for a particular preservation action.

In chapter 5, descriptions of a prototype transformation component that was tested by the PA/4 workpackage, capable of transforming graphics files from *.fig to *.dia format, are included. To demonstrate the difference between descriptions of tools and descriptions of pathways, these include a description of a pathway (the service 'FigToDia') as well as a description of the tool used for this transformation, Dia.

During the creation of these descriptions it became apparent that at some points, the properties of tools that are to be described in the language need to be defined in a less ambiguous way than what was used in the first version of the language. For instance, a property like 'creation date' for a software program could mean the date the original (first) version of the program was created, or the date that a specific version of the program was released. Where applicable, descriptions of these properties were changed to cover their intended use more precisely.

The model for describing the file formats that can be read and written by preservation action tools has been enhanced as well. While a detailed description of properties of file formats is not within the scope of this language, it should be possible to specify what formats can be used by a preservation action tool in an unambiguous way. The first version of the language only allowed for specification of a single file format, but in the current model the input and output for a preservation action tool can be a set of files as well to allow for the description of many-to-many conversions. Each of the files in such a set is referred to by means of a PRONOM Unique Identifier (PUID) which makes the exchange of information with the characterisation registry possible.

The way technical environments are described has also changed significantly since the first version. Host environments (technical environments required by software products) and target environments (technical environments offered by emulation software) now share the same set of properties in order to make automatic matching of these two entities possible. In the PA/5 workpackage the possibilities of automatically matching (descriptions of) host and target environments are currently investigated further, and it is still an open topic whether (an adaptation of) the set of properties defined in this version of PTDL will be used for this process or whether a

technical environment will be refered to by making use of a unique identifier. More background on the description of technical environments is given in paragraph 3.4.

Since new properties will be added to the tool description model during the project, elements have been added to make the language extensible in the future.

2.3 Design Overview

2.3.1 Relevant prior work

To prevent digital objects from becoming inaccessible due to software obsolescence, information on the programs that can be used to render these objects (or convert them into different formats) will have to be stored. Within the digital preservation community, a number of projects have worked on registries containing information on file formats, and some of these are designed to store information on the programs that work with those formats as well.

The PRONOM technical registry[5], developed at the National Archives, was designed for storing information on the technical environment a digital object needs to be rendered. The Global Digital File format registry (GDFR)[6], a joint effort between Harvard University Library and OCLC, also aims to describe not only the properties of file formats, but also the properties of software programs working with those formats. At the National Library of the Netherlands, an application called the Preservation Manager was designed with the same goal in mind: storing information on the environment needed to render a file format[7]. In the Australian PANIC project[8], an ontology in OWL/S was presented that is aimed specifically at the description of preservation actions. All these projects have relevancy for Planets as well, and the design of these systems has been taken into account in compiling the initial list of tool properties used in the PTDL language.

2.3.2 Methodology

In the development of the tool description language, we have started out by making an inventory of the kind of properties that current registries for tools to describe software programs. In particular, the National Archives' PRONOM system and the proposed data model for the GDFR were investigated and compared to ideas found in the design of the KB's Preservation Manager. Based on this, an initial list of properties was compiled.

Since the two types of tools that are to be described by the language (tools for objects and tools for environments) share a lot of properties, a design was chosen in which these two types of tools can be derived from a general preservation action tool description. A list of properties shared by both types of tools was created, and complemented by two separate lists of properties for both specific types of tools.

Members of the PA/2 workpackage, as well as participants in the PA/4 and PA/5 workpackages have been asked to comment on these lists, and to provide additional properties. Since the PA/4 and PA/5 workpackages are concerned with creating tools for objects and tools for environments, respectively, it was expected that experience from members of these workpackages could be used to enhance the initial model. In the discussions that followed, the need for describing less 'measurable' properties concerning the usability (as well as other aspects of the tools) became apparent. In paragraph 3.4 in the next chapter, more information is given on this subject. Exactly what properties will be used to support decisions by the planning system, and what elements will be used to provide additional information to humans, is something that will have to be decided in cooperation with members of the PP workpackage.

Another issue is caused by properties that can be measured in an objective way, but that depend on certain external factors (like the hardware the system is running on). For these, objective measures can only be taken in a controlled environment. Within the Planets project, the Testbed environment could be used to obtain measurements in such a controlled environment. Whether

these dependant measurements will be used in the automated selection of tools, or merely to provide additional information to the user has to be decided in cooperation with the PP workpackage as well.

To ensure that the currently defined set of properties is sufficient for accurately describing preservation action tools, an empirical method is used. The tools that are analysed by the PA/4 and PA/5 working groups are being described using this model (an example of such a description, of a conversion tool called 'Dia', is included in chapter 5). This process will have to continue in the coming months, and experience gathered in the creation of these descriptions is used to refine the language. Tools that will be described to validate and enhance the current model include the emulator that's being developed within the PA/5 workpackage, as well as other prototypes of transformation services that are tested by PA/4 (currently, these include the ImageMagick program for conversion of various bitmapped file formats and a tool for conversion of Microsoft Word documents to OpenXML format).

The initial design of the language that was presented in the first version of this document has been adapted to follow the data model that the PA/3 workpackage started to develop in march 2007. Tests with this registry and the planning system that´s being developed in the PP/4 workpackage, will have to show if the current set of properties that's defined under the cluster of properties called 'access' will be sufficient to allow for the automatic invocation of the various types of software stored in this registry.

2.4 Context & Dependencies

The tool description model and corresponding language developed within this workpackage are meant to provide a method for describing preservation action tools and their context of use. The descriptions made in this way can be used to perform a gap analysis (to identify missing tools) and to automatically select tools from a registry. The development of this model is dependent on several specific parts of the Planets system, thus on other workpackages within the Planets project. Many of these parts have not been developed at the time of writing of this document. It is important that the following dependencies will be explored in subsequent iterations of this document:

- **PC/2 Characterisation of Digital Objects**
  The evaluation of the functions and appropriate use of specific preservation action tools depends on how well significant properties of source and target files are preserved by the tool. The PC/2-D1 and D2 are concerned with a pair of XML-based extraction and description languages for digital objects (XCDL/XCEL). To prevent unnecessary overlap in work, the PA/2 workpackage group would like to discuss and define whether or not detailed description of significant properties of digital objects should be part of the generic description model for PA tools.
- **TB Testbed**
  The Planets Testbed will provide an environment that will enable the benchmark evaluation of preservation action tools. The experiments run on the Testbed will be the source for most of the evaluation information. The PA/2 workpackage group and the Testbed sub-project need to discuss the evaluation criteria that should be incorporated in the generic tool description.
- **PP/4 Preservation Planning Tools**
  As mentioned in the introduction to this document, development of the PTDL language is heavily dependent on the planning tool in the Planets system. The PP/4 and PA/2 workpackage groups must discuss and define the communication from and to the Planning Tool (e.g. what is the form and content of requests made by the planning tool, and what is the expected form and content of the answers it can process?).
- **PA/3 Preservation Action Tools Registry**
  The other important part in the Planets system that will make use of the PTDL language is the PA tools registry. The first version of the language has been adapted to follow the data model for the PA/3 registry as presented in the PA/3-D1 deliverable. Since that model is still bound to changes, further development of the registry will have to be monitored and

changes that influence the tool description language will have to be implemented on a regular basis.

- **PA/4 and PA/5: Preservation Action Tool Development**
  Although the DoW states that the PA/2 work package strategy development is guiding the tool developing work packages, the PA/2 work package group would like to stress the importance of reciprocal exchange of ideas among the PA/2, PA/4 and PA/5 work packages. The PA/4 and PA/5 work packages will gain (hands-on) experience in analysis of tools and tool building during the Planets project. In the current iteration of this document, some experiences of PA/4 and PA/5 workpackage members have been incorporated in the tool description model. In subsequent iterations, this process needs to continue.

2.5 Open topics

In the development of the tool description language presented in this document, a number of issues were raised that have to be taken into consideration in the further development of the language. This paragraph gives an overview of these topics.

- **PC/3 Preservation Characterisation Tools**
  For now, the PTDL language describes PA tools. However, the PC sub-project needs to describe characterisation tools used in the Planets project, and could make use of (an extended version of) the PTDL language as well. For this, a new 'CharacterisationTool' element (to be derived from the PreservationTool element) could be added. In this iteration of the language, this issue has not been addressed.
- **RDF/OWL-S compatibility**
  In the PANIC system, described in Jane Hunter's article [8], an ontology in OWL-S[4] is presented that's primarily aimed at the description of software tools for preservation. In this document, we present a tool description language based on XML-Schema, but it is not (yet) based on RDF or OWL-S. In other words, the language described in this document can be used as a format for the exchange of messages between a tools registry and a preservation planning system. However, it is not a knowledge representation that might be shared between these two parts of the system. In the next iteration of this language, compatibility with RDF and OWL-S will be considered.
- **Extensibility**
  It will clearly be desirable to add additional elements during the project. Also, with the current schema, the order of elements is fixed in a lot of places. This will make the parsing process easier, but puts additional constraints on the planning system and the tools registry. Future development of these systems will determine whether the current model will be sufficient or changes will have to be made.
- **Shared use of schemas**
  Since more XML-based languages will be used in the Planets project, the shared use of schema definitions would be desirable. During the project it should become clear what parts of schema definitions can be shared by multiple languages.
- **Subjective measurements**
  Some of the properties defined in the current model, especially in the 'evaluation' section, are very subjective in nature. In the current version of the language, the value for these properties can be chosen from a fixed set in order to enable automated processing. Whether or not these subjective measurements will be used in the automated selection of preservation action tools has to be decided in cooperation with the PP workpackage.
- **Description of technical environments**
  In the version of PTDL that's presented in this document, the description of a technical environment that's required by a software product (a host environment) or offered by a tool for environments (a target environment) has a fairly high amount of detail. The PA/5

workpackage is currently investigating the possibilities of automatically matching target environments (offered by emulation software) with host environments required by other software products (applications or operating systems). Whether an adaptation of the current, 'detailed', model will be used when referring to technical environments, or whether a set of standard environments that can be referenced by a unique ID will be used, is for now an open topic. More background information on this subject can be found in paragraph 2.5 of the document named 'Starting point for registration and discovery of PA/5 tools', available on the Planets Wiki at http://www.planets-project.eu/private/planets-ftp/WP_PA/PA5/PA5_startingpoints_registry_invocation_v0.2.doc

3 Language overview

3.1 Introduction

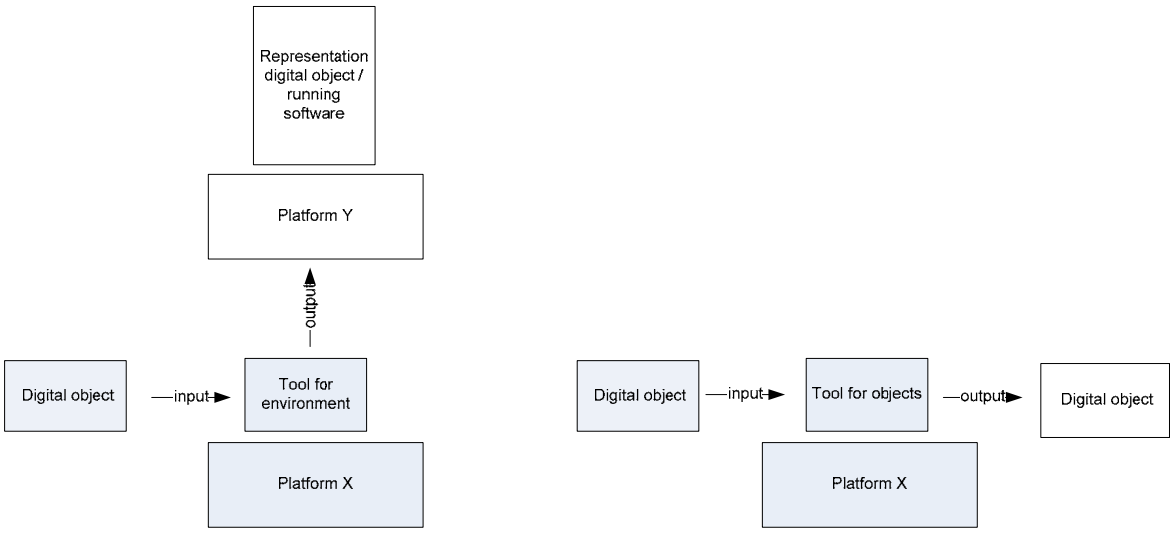
In this chapter, background information is given on the design decisions that were taken in the development of the PTDL language. Information on the methodology used in designing the language can be found in chapter 2.

3.2 Tool similarities and differences

The goal of the language developed within this workpackage is to provide a mechanism for the description of preservation action tools. These can be tools for objects or tools for environments. Tools for environments and tools for objects share the same objective: to provide long-term access to digital objects. However, the process for meeting that objective is quite different for these two types of tools.

Tools for environments provide an (emulated) environment other than the current environment in order to represent a digital object in an obsolete format. Tools for objects convert the object in an obsolete format into an object in a current format that can be rendered on the current environment. One of the most obvious differences between tools for objects and tools for environments is the 'output' of the preservation action process. The process of a tool for environments results in a representation of the digital object in the memory of the computer. For a tool for objects, the process result is a new or changed digital object. Despite these differences, the analysis of the properties of the two types of preservation action tools indicated that quite a few properties are shared by both types of tools. For example both types of tools use a digital object (the object to be rendered or converted) as input (see paragraph 3.4 for some additional notes on the input files for emulation software). Therefore, the generic description model contains a distinction between general properties (those that are relevant for tools for objects and for tools for environments) and specific properties (specific to tools for objects and tools for environments).

The diagram shows the shared and specific parts of both types of preservation action tools in a visual manner.



3.3 Pathways: one function, (possibly) multiple tools

Some tools offer the functionality to perform a number of different preservation actions. For instance, a conversion tool might be able to convert between a number of different bitmap formats, or an emulation tool might be able to offer a couple of different emulated platforms. From the viewpoint of a preservation planning tool, such a tool can be seen as a number of separate tools. That's why they would be described as such in the PTDL language as well. In other words, what we describe in a 'pathway' in this language, are tools put to a specific use. This is the reason why every described tool in a pathway should have only one input format, and tools for environments can, when seen as part of a pathway, only provide one emulated environment. Tools that offer multiple functionalities should be described multiple times in this model. In this way, an unambiguous description of a preservation action can be given.

Likewise, several tools might be needed to perform one preservation action. There might be no tool available to convert a WordPerfect document directly to PDF/A, for example, while this conversion can be accomplished by first using one tool to migrate the WordPerfect document into an intermediate format and then using another tool to convert that format to PDF/A. A similar situation with tools for environments would arise when one emulator runs on another emulator.

For the preservation planner such a set of tools can be considered to be one tool. In this document, the term 'pathway' is used to refer to a set of one or more preservation action tools that are being used to perform a preservation action.

Since the language should also be usable in cases when the user asks information about a specific tool, the number of input and output formats for a tool is not restricted in the schema. In this way the language can be used to supply information on a tool as part of a pathway (with one input and (possibly) one output format) or on a tool as a separate entity (with, possibly, multiple in- and output formats).

3.4 Technical environments

References to technical environments are used in the description of an environment that's required by a tool in order to run (a 'host environment'), as well as in describing environments that are 'offered' by tools for environments (a 'target environment'). In the current version of the PTDL language these environments share the same set of properties like 'processor speed', 'amount of storage space', etcetera (the only difference is the possibility to specify a 'recommended' and a 'minimum' value for most of the properties of a host environment, while for target environments only the 'recommended' values are used). The PA/5 workpackage is currently investigating the possibilities of automatically matching such environments, and of starting an emulation process with a disk image that contains a specific operating system and specific application software (In contrast to tools for objects or applications that render a file, emulators usually require an operating system and an application to be started before an object can rendered). Whether an adaptation of the current model or a more abstract representation of technical environments (possibly consisting of a number of standard environments with unique id numbers) will be used is currently an open topic. More background information on this subject can be found in paragraph 2.5 of the document named 'Starting point for registration and discovery of PA/5 tools' [11].

3.5 Subjective measurements

From the discussions held within the PA/2 working group, it showed that in some cases it can be useful to include properties with a more subjective nature in the description of a tool. For instance, in a review of preservation tools presented in *RLG DigiNews* [9], 'Difficulty level for configuration' is used as a criterion for evaluation. Also, the 'readability of the source code' was mentioned as an example of a significant property of preservation action tools. While these are certainly useful properties to consider when selecting from a number of tools, it's hard to provide a mechanism for describing such features in an objective way. In these cases, a choice from a fixed set of values is provided in order to make possible the automated processing of the tool descriptions. However, whether these properties will be used in the automated selection of tools or will only be used to

provide extra information to a human being will have to be decided in cooperation with the PP (Preservation Planning) workpackage.
In some cases, attempts have been made to provide a range of values that allows for a less subjective measure: The aforementioned 'difficulty level', for instance, was initially defined as a value ranging from 'very easy' to 'very hard', but this has been changed to a more meaningful value called 'skills level' later, ranging from 'general computer user' to 'computer programmer'.

3.6 Dependant measurements

Apart from values that are dependant on the opinions of human beings, like the ones mentioned in the previous paragraph, some measurements are dependant on other external factors. A property like the 'speed of operation' of a program depends on the hardware that the program runs on and doesn't make much sense outside of this context. Within the Planets project, the Testbed workpackage will provide an environment that enables the benchmark evaluation of such properties. Whether or not these properties will be used in the automated selection of tools will have to be decided in cooperation with the PP workpackage as well.

3.7 Overview of properties

The overview in this paragraph presents a list of properties that can be used to describe preservation action tools. A subdivision is made into **general** properties (*What kind of tool is this?*), information about the **developer** of the tool (*Who made it?*), about the **access** to the tool *(Where can I find it or how to use it?)*, **evaluation** *(What are other people's experiences with the program?)*, the **intellectual property rights** *(Under what circumstances am I allowed to use it?)* and the **dependencies** (*What is required to run this tool?*).

Note: Some of the data types mentioned in this table (and in the accompanying schema presented in paragraph 4.2) still have to be decided on. Also, developments in the PP workpackage should provide information on what properties will be used in decision-making by the planning system, and what properties are merely used to provide extra information to human beings.

Table 1.1: properties

Property	Description	Data type	Example / Comments
general			
Name	The name of the tool	String	ImageMagick
Version	The version number	String	0.8, 1.4 beta
Description	A short, human-readable description of the tool that gives a clear indication of its functionality. This should include the file type(s) that the program can work with, along with the version number. If this tool description describes the tool in general it can include all the file types that the program can work with, but for tools that are part of a pathway the description should make clear what specific conversion the tool is used for.	String	This is a webservice that can be used to convert TIFF 4.0 images to TIFF 5.0, making use of the ImageMagick program.
ReleaseDate	The date that the version of the program that this tool	Date	2003-02-08

Property	Description	Data type	Example / Comments
	description is about was originally released.		
Location	An URL of the location where the program can be used / downloaded. For web services, this would be the location of the service itself. For stand-alone programs this could be a location where the program can be downloaded.	URI	http://docmorph.nlm.nih.gov/docmorph/setup1.exe
InstallGuide	Includes information on the installation guide of the program (if available)	Documentation	
UserGuide	Includes information on the user guide of the program (if available)	Documentation	
ProgrammingLanguage	The name of the programming language that the program was written in.	String	C++. (An enumeration could be used as well, to make automatic selection easier, but this property seems to be only of interest to humans)
developer			
Name	Name of the person or company that released the program	String	ImageMagick
Type	Used to specify whether the developer is an organisation or an individual	Enumeration	
Website	URL of the website of the company or person that released the program	URI	http://www.imagemagick.org
Mail	Contact information for the person or company that created the software	URI	info@imagemagick.org
SupportUntil	The date that support for the program was, or is due to be, officially withdrawn.	Date	1997-01-01
Access			
InputFormat	The file type(s) that the program can read	FileSet	In case the tool description describes a tool that's part of a pathway, only the format(s) used in this pathway should be included here. Otherwise, all file types that the program can read or write can be included.
InputMaxSize	The maximum file size that the program can handle	Enumeration (memory space)	4 MB
InterfaceType	Indicates the type of interface of the program	Enumeration (interface type)	GUI, command line, web service
UILanguage	The (natural) language(s) that the user interface of the program supports	Language	En, de, fr
ProgrammingInterface	Specifies whether the program offers an Application Programming Interface so that it's functionality can be used by other software programs.	Boolean	Yes, no
evaluation			
Speed	Gives an indication of the speed	Integer	Speed measurements should

Property	Description	Data type	Example / Comments
	of operation of the program, based on previous experience		be seen in context of a hardware environment. Within the Planets project the Testbed provides an environment that allows measurements like these to be taken. A value measured in 'seconds per processed kilobyte' could be used here, although this will be dependant on the nature and complexity of the processed objects.
Stability	Gives an indication of the stability of the program, based on previous experience	Enumeration	unstable, very stable
Accuracy	Gives an indication of the accuracy of the program (e.g. how well are significant properties preserved after conversion, or how well can you play an emulated game), based on previous experience	Enumeration	inaccurate, accurate
DifficultyDownload	Gives an indication of how hard it is to obtain the program	Enumeration (skills level)	General user
TimeDownload	Gives an indication of the time needed for downloading the program	Enumeration (duration)	Duration. This property is dependent on the speed of the connection. Within the Planets project, results from tests within the Testbed environment should be used in order to make comparisons possible.
DifficultyInstall	Gives an indication of the level of technical competence a user should have in order to install the program	Enumeration (skills level)	General user, System administrator
TimeInstall	Gives an indication of the amount of time it took to install the program	Enumeration (duration)	Duration
DifficultyConfig	Gives an indication of the level of technical expertise a user would need to correctly configure the program	Enumeration (skills level)	General user, System administrator
TimeConfig	Gives an indication of the time spend configuring the program before it was ready for use	Enumeration (duration)	Duration
DifficultyUse	Gives an indication on the level of technical expertise needed to operate the program	Enumeration (skills level)	General user, System administrator
…			Since more specific properties will be used in the evaluations of specific tools, more elements can be added here
IPR			
Type	Specifies the type of license of the software.	Enumeration	Free software, proprietary, GPL, GPL incompatible,

Property	Description	Data type	Example / Comments
			Copylefted, Pay Licensed Closed Source Note: E.g. software can be free software, but GPL incompatible
Description	Can be used to give a small description of the type of license	String	
Owner	Specifies the person or organisation that owns the license	Agent	
dependencies			
HostEnvironment	The technical environment that the tool needs	TechnicalEnvironment	As the amount of detail in the descriptions of technical environments is currently an open topic, the list of properties that's used to describe a technical environment is bound to change.

In the following table, properties that are only relevant for tools for objects are listed:

Table 1.2: specific properties for tools for objects

Property	Description	Data type	Example
Tools for objects			
OutputFormat	The file type(s) that the program can write	FileSet	PDF, JPEG
BatchProcessing	Specifies whether this program can handle multiple files in a batch	Boolean	Yes
OutputMultiple	Specifies whether this tool will output multiple files in different formats when performing this action	Boolean	Yes or no, to specify if multiple files are outputted for a single input file. For example: TIFF > BMP + TXT for metadata

In the following table, properties that are only relevant for tools for environments are listed. As the amount of detail in the descriptions of technical environments is currently an open topic, the list of properties that's used to describe a technical environment is bound to change.

Table 1.3: specific properties for tools for environments

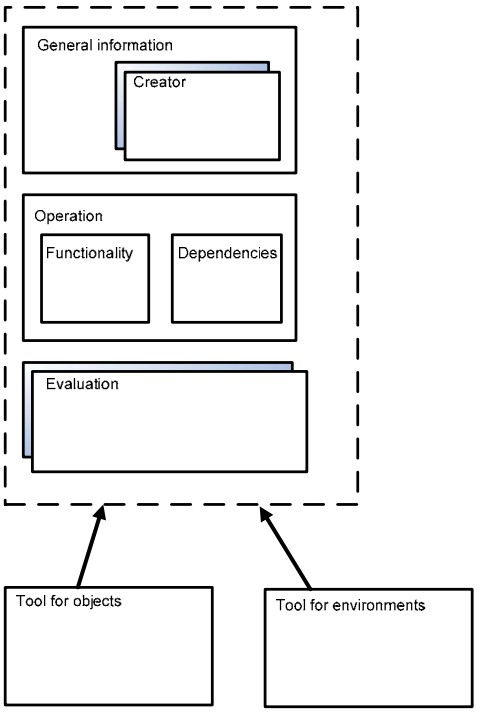
Property	Description	Data type	Example
Tools for environments			
TargetEnvironment	The technical environment that this tool offers	TechnicalEnvironment	As the amount of detail in the descriptions of technical environments is currently an open topic, the list of properties that's used to describe a technical environment is bound to change.

4 PTDL: An XML-based tool description language

4.1 Introduction

In this chapter, the schema for PTDL is presented. In it's current version, the schema consists of only one xsd file. This might change in later versions of the language, when more elements will be added or inclusion of data types from other schemas becomes necessary.

4.2 Technical decisions



The tools that are to be described are divided into two groups; tools for objects and tools for environments. Because these two types of programs differ in functionality but share a lot of properties as well, the descriptions for these specific types of tools are derived from a generic tool description. Apart from the properties that both types of programs share, a description for a tool for environments also contains information on the emulated platform, and a tool for objects contains information on the types of files that the program can output. File formats for input can be given both for tools for environments and tools for objects, since a program that emulates a hardware environment can be said to read a file as well when it opens a program. In contrast to tools for objects, these programs don't turn such files into other files but present to the user a representation of the file's contents.

XML Schema is used for the definition of the language [4]. At the moment, the systems that will use this language to exchange information are still under development, so some design decisions are left open. For instance, some data types are still not defined to the level of detail that they probably should be. Mostly this is because they will have to communicate with other parts of the Planets system. The level of detail in descriptions of technical environments is an example of this. Also, some of the enumerated lists might have to be extended in the future (e.g. lists of Storage devices or Input devices). We chose to use enumerated lists instead of just strings in a lot of cases, because this would make it easier for an automated planning system to use the data.

4.3 Elements

This paragraph contains a more detailed description of the complex elements in the schema. Here, only the elements that are made out of other elements (so called complexTypes) are described. Elements that are made up of the basic data types that are already defined in the W3C XML Schema specification[2] are not included in these descriptions.

The root element <ptdl>	Required: Yes Repeatable: No
All PTDL documents start with the <ptdl> root element that wraps at least one (but possibly more) tool descriptions. Such a tool description consists of a <ObjectTool> element or an <EnvironmentTool> element, both of which are complex types that are derived from the <PreservationTool> element. They can have an attribute called 'id' that is used to store a unique identification number for the tool. In the schema this id has not been implemented yet, because it's not clear yet how interfacing with the tools registry will take place.	

The <Pathway> element	Required: No Repeatable: Yes
This is a repeatable element that can be used to describe a pathway. A pathway can consist of a number of tools, and can also include a description and an evaluation. Every pathway also includes a property called 'status' that can be used to specify if the pathway is approved, under construction, etcetera.	

The <ObjectTool> element	Required: No Repeatable: Yes
The <ObjectTool> element can be used to describe migration tools (tools for objects). Like any tool description, The <ObjectTool> element can have the three child elements <GeneralInfo>, <HostEnvironment> and <Source>. In addition, a number of output formats can be specified for any migration tool, and another element is added that can be used to tell whether the tool can handle multiple files in a batch.	

The <EnvironmentTool> element	Required: No Repeatable: Yes
The <EnvironmentTool> element is similar in function to the <ObjectTool> element, but describes an emulation or virtualisation tool instead. Like the <ObjectTool> element, this element also contains the three child elements <GeneralInfo>, <HostEnvironment> and <Source>, but there is another element called <TargetEnvironment> to store information on the kind of platform this tool can emulate (or virtualise).	

The <GeneralInfo> element	Required: Yes Repeatable: No
The <GeneralInfo> element is used to store information on the creator, the license and a couple of other features that are shared by all software tools. As child elements it can contain elements like <Name>, <Version>, and <Description>, as well as any number of <Creator> elements.	

The <Documentation> element	Required: No Repeatable: Yes
This is a repeatable element that can be used to specify information on a piece of documentation that describes a software product.	

The <Agent> element	Required: No Repeatable: Yes
This is a repeatable element that can be used to specify information on a person or organisation that created a piece of software or documentation, or supplied a piece of information. It wraps a couple of child elements that contain information like the name and contact information.	

The <Evaluation> element	Required: No Repeatable: Yes
The <Evaluation> element is a repeatable element that contains information on the experiences a reviewer had when using the described pathway. For every pathway one evaluation can be given, that should be based on tests run in the Planets testbed environment. These values are of a subjective nature, and a 'skills level' ranging from a 'general computer user' to 'computer programmer' can be associated with a number of aspects of the program.	

The <HostEnvironment> element	Required: Yes Repeatable: Yes
This element is used to describe the external dependencies of the program, or what one needs in order to run it. It makes use of the same set of properties as a 'target environment' for tools for environments, but for most properties a ´minimum´ and ´recommended´ value can be specified.	

The <Provenance> element	Required: Yes Repeatable: No
This element is used in the description of provenance information for most entities. It is a required element for these entities, and can be used to specify the source of the information that´s stored in the registry.	

The <TargetEnvironment> element	Required: Yes Repeatable: No
This element is used in the description the platform (environment) that an emulator or virtualisation tool offers. It uses the same set of properties that are also used to describe a 'host environment' (an environment that's required by a tool), but in this case only the 'recommended' properties are used to specify the characteristics of the emulated environment.	

4.4 Schema

This is the schema definition of the tool description language. Currently, it's contained in one document, but when extensions will be made and this definition grows larger, the definition for a general tool could be separated from the descriptions for tools for objects and tools for environments, resulting in three separate documents.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Schema for the description of preservation action tools and their context of use. This can be used to
describe
            seperate preservation action tools as well as tools put to a specific use (tools as part of a pathway)
        </xsd:documentation>
    </xsd:annotation>
    <!-- Enumerations used by both migration and emulation tools -->
    <xsd:simpleType name="MemoryUnit">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="byte"/>
            <xsd:enumeration value="kilobyte"/>
            <xsd:enumeration value="megabyte"/>
            <xsd:enumeration value="gigabyte"/>
            <xsd:enumeration value="terabyte"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Frequency">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Hertz"/>
            <xsd:enumeration value="Kilohertz"/>
            <xsd:enumeration value="Megahertz"/>
            <xsd:enumeration value="Gigahertz"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="InterfaceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="GUI"/>
            <xsd:enumeration value="Command line"/>
            <xsd:enumeration value="Webservice"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Speed">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Very slow"/>
            <xsd:enumeration value="Slow"/>
            <xsd:enumeration value="Average"/>
            <xsd:enumeration value="Fast"/>
            <xsd:enumeration value="Very fast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Stability">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Very unstable"/>
            <xsd:enumeration value="Unstable"/>
            <xsd:enumeration value="Stable"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Accuracy">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Very inaccurate"/>
            <xsd:enumeration value="Inaccurate"/>
            <xsd:enumeration value="Accurate"/>
            <xsd:enumeration value="Very accurate"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="DifficultyLevel">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Very easy"/>
```

```
            <xsd:enumeration value="Easy"/>
            <xsd:enumeration value="Average"/>
            <xsd:enumeration value="Hard"/>
            <xsd:enumeration value="Very hard"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="IPR">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="GPL"/>
            <xsd:enumeration value="Proprietary commercial software"/>
            <xsd:enumeration value="Shareware"/>
            <xsd:enumeration value="Non-copylefted free software"/>
            <xsd:enumeration value="Open Source software"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- Enumerations for emulators -->
    <xsd:simpleType name="BitDepth">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="1bit"/>
            <xsd:enumeration value="2bit"/>
            <xsd:enumeration value="4bit"/>
            <xsd:enumeration value="8bit"/>
            <xsd:enumeration value="16bit"/>
            <xsd:enumeration value="24bit"/>
            <xsd:enumeration value="32bit"/>
            <xsd:enumeration value="64bit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="AudioSampleRate">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="22050khz"/>
            <xsd:enumeration value="44100khz"/>
            <xsd:enumeration value="11025khz"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="DataExtraction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="images"/>
            <xsd:enumeration value="binary"/>
            <xsd:enumeration value="text, images"/>
            <xsd:enumeration value="images, binary"/>
            <xsd:enumeration value="text, images and binary"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="InputDevice">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Mouse"/>
            <xsd:enumeration value="Keyboard"/>
            <xsd:enumeration value="Joystick"/>
            <xsd:enumeration value="Trackball"/>
            <xsd:enumeration value="Gamepad"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="StorageDevice">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="FloppyDisk"/>
            <xsd:enumeration value="CDRom"/>
            <xsd:enumeration value="Tape"/>
            <xsd:enumeration value="Cartridge"/>
            <xsd:enumeration value="USBStorageDevice"/>
            <xsd:enumeration value="Harddisk"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--
    ____________________________________________________________________
    |                                                                  |
    |                                                                  |
    |                                                                  |
    |              Start of element declarations                       |
```

```
    |                                                                                |
    |                                                                                |
    |________________________________________________________________________________|

    -->
    <xsd:element name="ptdl">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                The main element that can be used to wrap the descriptions of sets of pathways or sets of tools
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Pathway" type="Pathway" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="MigrationTool" type="MigrationTool" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element name="EmulationTool" type="EmulationTool" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="version" type="xsd:string" use="required"/>
        </xsd:complexType>
    </xsd:element>
    <!-- Definition for pathway -->
    <xsd:complexType name="Pathway">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element is used to describe a collection of one or more tools that can be used to perform a
preservation action. Such a set of tools is called a pathway
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Evaluation" type="Evaluation" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
            <xsd:element name="EmulationTool" type="EmulationTool" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="MigrationTool" type="MigrationTool" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="Source" type="Provenance"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:integer" use="required"/>
        <xsd:attribute name="status" type="xsd:string" use="required"/>
    </xsd:complexType>
    <!-- Definition for a file -->
    <xsd:complexType name="File">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element describes a single file.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:any namespace="##local" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            <!-- 'any'-element added for extensibility -->
        </xsd:sequence>
        <xsd:attribute name="type" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:pattern value="(x-)?(fmt)/([0-9]*[a-z]*)+"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="optional" type="xsd:string" use="required"/>
        <!-- Attribute for specifying the file type in the form of a PRONOM PUID. Regexp allows any PUID that
contains an 'fmt'-token -->
    </xsd:complexType>
    <!-- Definition for a set of files -->
    <xsd:complexType name="FileSet">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element describes a set of files. It's used to specify the set of files a tool can read or write.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="File" type="File" minOccurs="0" maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for an agent -->
    <xsd:complexType name="Agent">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element is used to describe the creator of a software program. This
                can be either an individual person or a company.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Name" type="xsd:string"/>
            <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Address" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Website" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Mail" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Telephone" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Support" type="xsd:boolean" minOccurs="0"/>
            <xsd:element name="Source" type="Provenance" minOccurs="0"/>
            <xsd:any namespace="##local" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for provenance information -->
    <xsd:complexType name="Provenance">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element is used to keep track of the source of the information in the registry. It can be used to
describe the person or organisation that provided the information in the first place, as well as the last person or
organisation that modified the information. In addition, the date the information was submitted and modified for the
last time can be stored here.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="SuppliedBy" type="Agent"/>
            <xsd:element name="SuppliedAt" type="xsd:dateTime"/>
            <xsd:element name="LastModifiedBy" type="Agent"/>
            <xsd:element name="LastModifiedAt" type="xsd:dateTime"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for documentation -->
    <xsd:complexType name="Documentation">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element is used to store information on the documentation of a software product or hardware
component
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Creator" type="Agent" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Location" type="xsd:anyURI"/>
            <xsd:element name="ReleaseDate" type="xsd:date"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for memoryspace -->
    <xsd:complexType name="MemorySpace">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element is used to describe a certain memory size,
                measured in bytes, kilobytes, megabytes, or gigabytes
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Amount" type="xsd:integer"/>
            <xsd:element name="Unit" type="MemoryUnit"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for processorspeed -->
    <xsd:complexType name="ProcessorSpeed">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
```

```
                This element describes the speed of a microprocessor. It's used to describe the
                speed of the processor that a program needs in order to run (under dependencies) and
                to specify the speed of an emulated processor in an emulator
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Amount" type="xsd:integer"/>
            <xsd:element name="Unit" type="Frequency"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for general information for a tool. -->
    <xsd:complexType name="GeneralInfo">
        <xsd:sequence>
            <xsd:element name="Name" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="Version" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Location" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="InstallGuide" type="Documentation" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="UserGuide" type="Documentation" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ProgrammingLanguage" type="xsd:string" minOccurs="0"/>
            <xsd:element name="CreationDate" type="xsd:date" minOccurs="0"/>
            <xsd:element name="Developer" type="Agent" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="InputFormat" type="FileSet" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="InputMaxSize" type="MemorySpace" minOccurs="0"/>
            <xsd:element name="InterfaceType" type="InterfaceType" maxOccurs="3"/>
            <xsd:element name="UILanguage" type="xsd:language" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ProgrammingInterface" type="xsd:boolean" minOccurs="0"/>
            <xsd:element name="License" type="IPR" minOccurs="0"/>
            <xsd:element name="Price" type="xsd:decimal" minOccurs="0"/>
            <xsd:any namespace="##local" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- Definition for evaluation -->
    <xsd:complexType name="Evaluation">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element can contain information on the experiences
                people had when using a software program
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Speed" type="Speed"/>
            <xsd:element name="Stability" type="Stability"/>
            <xsd:element name="Accuracy" type="Accuracy"/>
            <xsd:element name="DifficultyDownload" type="DifficultyLevel"/>
            <xsd:element name="DifficultyInstall" type="DifficultyLevel"/>
            <xsd:element name="DifficultyConfig" type="DifficultyLevel"/>
            <xsd:element name="DifficultyUse" type="DifficultyLevel"/>
            <xsd:element name="DifficultyAverage" type="DifficultyLevel"/>
            <xsd:any namespace="##local" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="Source" type="Provenance"/>
            <!-- Element for extensibility -->
        </xsd:sequence>
        <xsd:attribute name="reviewer" type="xsd:string" use="required"/>
    </xsd:complexType>
    <!-- Definition for a technical environment -->
    <xsd:complexType name="TechnicalEnvironment">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This element describes a technical environment. This can be used to describe a technical
environment that's required by
                a software product (a 'host environment') or a technical environment that's offered by an emulator (a
'target environment').
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="OperatingSystem" type="sfwPUID" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ServicePack" type="xsd:string" minOccurs="0"/>
            <xsd:element name="ProcessorType" type="xsd:string" minOccurs="0"/>
            <xsd:element name="ProcessorSpeedMinimum" type="ProcessorSpeed" minOccurs="0"/>
            <xsd:element name="ProcessorSpeedRecommended" type="ProcessorSpeed" minOccurs="0"/>
```

```
            <xsd:element name="MemoryMinimum" type="MemorySpace" minOccurs="0"/>
            <xsd:element name="MemoryRecommended" type="MemorySpace"/>
            <xsd:element name="StorageSpaceMinimum" type="MemorySpace" minOccurs="0"/>
            <xsd:element name="StorageSpaceRecommended" type="MemorySpace"/>
            <xsd:element name="ScreenResolutionHMinimum" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="ScreenResolutionHRecommended" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="ScreenResolutionVMinimum" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="ScreenResolutionVRecommended" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="ScreenColourDepthMinimum" type="BitDepth" minOccurs="0"/>
            <xsd:element name="ScreenColourDepthRecommended" type="BitDepth" minOccurs="0"/>
            <xsd:element name="AudioBitDepthMinimum" type="BitDepth" minOccurs="0"/>
            <xsd:element name="AudioBitDepthRecommended" type="BitDepth" minOccurs="0"/>
            <xsd:element name="AudioChannelsMinimum" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="AudioChannelsRecommended" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="AudioSampleRate" type="AudioSampleRate" minOccurs="0"/>
            <xsd:element name="InputDevice" type="InputDevice" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="StorageDevice" type="StorageDevice" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="NetworkDevice" type="xsd:boolean" minOccurs="0"/>
            <xsd:element name="DataExtraction" type="DataExtraction" minOccurs="0"/>
            <xsd:any namespace="##local" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="sfwPUID">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="(x-)?(sfw)/([0-9]*[a-z]*)+"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- Definition for a software product -->
    <xsd:complexType name="SoftwareProduct">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                Basic definition for a software product. This can be used to describe preservation action tools,
                but also other software products such as operating systems. The MigrationTool and EmulationTool
                elements are derived from this element
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="GeneralInfo" type="GeneralInfo"/>
            <xsd:element name="HostEnvironment" type="TechnicalEnvironment" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="Source" type="Provenance"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="sfwPUID"/>
    </xsd:complexType>
    <!-- Definition for emulation tool -->
    <xsd:complexType name="EmulationTool">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                Definition for an emulation tool, derived from PreservationTool. This element is
                used for describing 'tools for environments'. These are tools that can be used
                to mimic a certain technical environment by means of emulation or virtualisation
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
            <xsd:extension base="SoftwareProduct">
                <xsd:sequence>
                    <xsd:element name="TargetEnvironment" type="TechnicalEnvironment"
maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- Definition for migration tool -->
    <xsd:complexType name="MigrationTool">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                Definition for a migration tool; derived from PreservationTool. This element is
                used for describing tools that can convert files from one type to another.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
```

```
            <xsd:extension base="SoftwareProduct">
                <xsd:sequence>
                    <xsd:element name="OutputFormat" type="FileSet" maxOccurs="unbounded"/>
                    <xsd:element name="BatchProcessing" type="xsd:boolean"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>
```

5 XML instances & test results

In this chapter, two examples of instances of a PTDL document are presented. The first instance describes a tool for drawing diagrams called Dia. This tool was chosen since it has been used as a prototype for a conversion service in the PA/4 workpackage (See the document titled 'Tool and service prototypes for internal evaluation'[10] for more information), so information on it's application was available. (The goal of these prototype tests was to gain experience with the implementation of wrapped tools as webservices, not to thoroughly test the tools themselves. The values for properties regarding the tool's evaluation should therefore not be taken seriously). Since this first instance describes a tool rather than a pathway, all input and output formats for the tool are listed, no evaluation is included and information on the required host environment is included in the description. Also, the type of interface is set to 'GUI' and the URL that's included under 'location' points to the place where the program can be (manually) downloaded.

The second instance describes a webservice that makes use of the same tool (Dia) for the conversion of the *.fig format to the *.dia format.

In cases where no known PUID was available for a file format or software product, an example has been used in these instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<ptdl xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Temp\xsdvalid\preservationtool.xsd" version="1.1">
    <MigrationTool id="x-sfw/381">
            <GeneralInfo>
                <Name>Dia</Name>
                <Version>0.96.1</Version>
                <Description>Dia is a diagram creation program released under the GPL license. It can load and
save diagrams to a custom XML format, and can export diagrams to a number of formats, including EPS, SVG,
XFIG, WMF and PNG.</Description>
                <Location>http://www.gnome.org/projects/dia/</Location>
                <UserGuide>
                    <Creator>
                        <Name>Kevin Breith</Name>
                    </Creator>
                    <Creator>
                        <Name>Henry House</Name>
                    </Creator>
                    <Creator>
                        <Name>Judith Samson</Name>
                    </Creator>
                    <Location>http://www.gnome.org/projects/dia/doc/dia-manual.pdf</Location>
                    <ReleaseDate>2000-10-06</ReleaseDate>
                </UserGuide>
                <CreationDate>2007-03-29</CreationDate>
                <Developer>
                    <Name>Alexander Larsson</Name>
                    <Website/>
                    <Mail>alla@lysator.liu.se</Mail>
                    <Support>false</Support>
                    <Source>
                        <SuppliedBy>
                            <Name>Administrator</Name>
                            <Website>http://www.institute.com/johndoe</Website>
                            <Mail>johndoe@institute.org</Mail>
                        </SuppliedBy>
                        <SuppliedAt>2002-05-30T09:00:00</SuppliedAt>
                        <LastModifiedBy>
                            <Name>AnotherAdministrator</Name>
                            <Website>http://www.institute.org/another</Website>
                            <Mail>another@institute.org</Mail>
                        </LastModifiedBy>
                        <LastModifiedAt>2002-05-30T09:30:00</LastModifiedAt>
                    </Source>
                </Developer>
```

```
            <InputFormat>
                <File type="x-fmt/381" optional="true"/>
                <File type="fmt/84" optional="true"/>
            </InputFormat>
            <InputMaxSize>
                <Amount>48</Amount>
                <Unit>megabyte</Unit>
            </InputMaxSize>
            <InterfaceType>GUI</InterfaceType>
            <UILanguage>en</UILanguage>
            <ProgrammingInterface>false</ProgrammingInterface>
            <License>GPL</License>
        </GeneralInfo>
        <HostEnvironment>
            <OperatingSystem>sfw/183</OperatingSystem>
            <OperatingSystem>x-sfw/37</OperatingSystem>
            <OperatingSystem>x-sfw/31</OperatingSystem>
            <OperatingSystem>x-sfw/9</OperatingSystem>
            <OperatingSystem>x-sfw/8</OperatingSystem>
            <MemoryMinimum>
                <Amount>256</Amount>
                <Unit>megabyte</Unit>
            </MemoryMinimum>
            <MemoryRecommended>
                <Amount>512</Amount>
                <Unit>megabyte</Unit>
            </MemoryRecommended>
            <StorageSpaceMinimum>
                <Amount>4</Amount>
                <Unit>megabyte</Unit>
            </StorageSpaceMinimum>
            <StorageSpaceRecommended>
                <Amount>12</Amount>
                <Unit>megabyte</Unit>
            </StorageSpaceRecommended>
            <ProcessorMinimum>
                <Amount>120</Amount>
                <Unit>Megahertz</Unit>
            </ProcessorMinimum>
            <ProcessorRecommended>
                <Amount>400</Amount>
                <Unit>Megahertz</Unit>
            </ProcessorRecommended>
        </HostEnvironment>
        <Source>
            <SuppliedBy>
                <Name>Administrator</Name>
                <Website>http://www.institute.com/johndoe</Website>
                <Mail>johndoe@institute.org</Mail>
            </SuppliedBy>
            <SuppliedAt>2002-05-30T09:00:00</SuppliedAt>
            <LastModifiedBy>
                <Name>AnotherAdministrator</Name>
                <Website>http://www.institute.org/another</Website>
                <Mail>another@institute.org</Mail>
            </LastModifiedBy>
            <LastModifiedAt>2002-05-30T09:30:00</LastModifiedAt>
        </Source>
        <OutputFormat>
            <File type="fmt/54" optional="false"/>
            <File type="x-fmt/540" optional="false"/>
        </OutputFormat>
        <BatchProcessing>false</BatchProcessing>
    </MigrationTool>
</ptdl>
```

In the following xml-instance the webservice ‘FigToDia’ is described in PTDL. It makes use of the tool ‘Dia’ for which a PTDL description is presented above, but in this case this tool is part of a pathway for the conversion of files in the *.fig format to files in the *.dia format. In this instance, only one input and output format are given for the service, and an evaluation is included. Also, the

values of some properties under ‘GeneralInfo’ differ from the example presented above, and no information on the host environment is included since this information is not relevant for a webservice.
Currently, the information on how to access a service or tool only includes a specification of the type of interface the tool uses and a location (URL). If the type of interface is ‘webservice’, than the URL under location can be used to point to a WSDL description of the service. Whether this is sufficient or a more detailed description of the method of tool invocation should be provided is a topic that needs further investigation.

```
<?xml version="1.0" encoding="UTF-8"?>
<ptdl xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="preservationtool.xsd" version="1.1">
    <Pathway id="3" status="approved">
        <Evaluation reviewer="221">
            <Speed>Fast</Speed>
            <Stability>Stable</Stability>
            <Accuracy>Accurate</Accuracy>
            <DifficultyDownload>Easy</DifficultyDownload>
            <DifficultyInstall>Easy</DifficultyInstall>
            <DifficultyConfig>Hard</DifficultyConfig>
            <DifficultyUse>Average</DifficultyUse>
            <DifficultyAverage>Average</DifficultyAverage>
            <Source>
                <SuppliedBy>
                    <Name>Administrator</Name>
                    <Website>http://www.institute.com/johndoe</Website>
                    <Mail>johndoe@institute.org</Mail>
                </SuppliedBy>
                <SuppliedAt>2002-05-30T09:00:00</SuppliedAt>
                <LastModifiedBy>
                    <Name>AnotherAdministrator</Name>
                    <Website>http://www.institute.org/another</Website>
                    <Mail>another@institute.org</Mail>
                </LastModifiedBy>
                <LastModifiedAt>2002-05-30T09:30:00</LastModifiedAt>
            </Source>
        </Evaluation>
        <MigrationTool id="x-sfw/381">
            <GeneralInfo>
                <Name>FigToDia</Name>
                <Version>0.96.1</Version>
                <Description>FigToDia is a webservice that can convert files from the *.fig format to *.dia format,
making use of the program called Dia.</Description>
                <Location>http://www.planets-project.eu/services/FigToDia/</Location>
                <ProgrammingLanguage>C</ProgrammingLanguage>
                <CreationDate>2007-03-29</CreationDate>
                <Developer>
                    <Name>Alexander Larsson</Name>
                    <Website/>
                    <Mail>alla@lysator.liu.se</Mail>
                    <Support>false</Support>
                    <Source>
                        <SuppliedBy>
                            <Name>Administrator</Name>
                            <Website>http://www.institute.com/johndoe</Website>
                            <Mail>johndoe@institute.org</Mail>
                        </SuppliedBy>
                        <SuppliedAt>2002-05-30T09:00:00</SuppliedAt>
                        <LastModifiedBy>
                            <Name>AnotherAdministrator</Name>
                            <Website>http://www.institute.org/another</Website>
                            <Mail>another@institute.org</Mail>
                        </LastModifiedBy>
                        <LastModifiedAt>2002-05-30T09:30:00</LastModifiedAt>
                    </Source>
                </Developer>
                <InputFormat>
                    <File type="x-fmt/381" optional="true"/>
                </InputFormat>
```

```
            <InputMaxSize>
                <Amount>48</Amount>
                <Unit>megabyte</Unit>
            </InputMaxSize>
            <InterfaceType>Webservice</InterfaceType>
            <License>GPL</License>
        </GeneralInfo>
        <Source>
            <SuppliedBy>
                <Name>Administrator</Name>
                <Website>http://www.institute.com/johndoe</Website>
                <Mail>johndoe@institute.org</Mail>
            </SuppliedBy>
            <SuppliedAt>2002-05-30T09:00:00</SuppliedAt>
            <LastModifiedBy>
                <Name>AnotherAdministrator</Name>
                <Website>http://www.institute.org/another</Website>
                <Mail>another@institute.org</Mail>
            </LastModifiedBy>
            <LastModifiedAt>2002-05-30T09:30:00</LastModifiedAt>
        </Source>
        <OutputFormat>
            <File type="fmt/54" optional="false"/>
        </OutputFormat>
        <BatchProcessing>false</BatchProcessing>
    </MigrationTool>
    <Source>
        <SuppliedBy>
            <Name>Administrator</Name>
            <Website>http://www.institute.com/johndoe</Website>
            <Mail>johndoe@institute.org</Mail>
        </SuppliedBy>
        <SuppliedAt>2002-05-30T09:00:00</SuppliedAt>
        <LastModifiedBy>
            <Name>AnotherAdministrator</Name>
            <Website>http://www.institute.org/another</Website>
            <Mail>another@institute.org</Mail>
        </LastModifiedBy>
        <LastModifiedAt>2002-05-30T09:30:00</LastModifiedAt>
    </Source>
  </Pathway>
</ptdl>
```

Generally, the current set of properties seems to be a good starting point for describing and comparing preservation action tools. However, a more thorough validation of this set of properties will have to take place by making more descriptions of a wider variety of tools. A number of areas in which the language needs further refinement are:

Evaluations
The model for describing evaluations of tools is not much more than a first proposal at the moment, and experiences gained in using the testbed environment in Planets will have to be used to enhance the current set of criteria.

Access
The way a tool can be accessed is modelled in a rather simple way in the current model. A distinction can be made between stand-alone tools and webservices, and a URL can be used to specify the location of the tool. Whether a more detailed description of the means by which a tool can be accessed should be used is a topic that needs further investigation.

Technical environments
As mentioned under ‘open topics’ (paragraph 2.5), the level of detail that should be used in the description of technical environments has not been decided on at the moment.

Also, since at the time this document was written the exact form and content of the information needed by the planning system is not yet known, changes can be expected when more information on this system comes available. Meanwhile more tests with the creation of tool descriptions will be used to make the generic description model fit for purpose.

References

[1] World Wide Web Consortium: Extensible Markup Language
http://www.w3.org/XML/

[2] World Wide Web Consortium: XML Schema: W3C recommendation, 28 October 2004.
http://www.w3.org/TR/xmlschema-0/

[3] World Wide Web Consortium: Resource Description Framework: W3C Recommendation, 10 February 2004.
http://www.w3.org/TR/rdf-schema/

[4] The OWL Services Coalition, OWL-S: Semantic Markup for Web services, DAML, 24 July 2004
http://www.daml.org/services/owl-s/1.1B/owl-s/owl-s.html

[5] Brown, Adrian (2005): 'PRONOM 4 Information Model',
http://www.nationalarchives.gov.uk/aboutapps/fileformat/pdf/pronom_4_info_model.pdf

[6] Data Model for the Global Digital Format Registry (GDFR)
http://hul.harvard.edu/gdfr/documents/GDFR-Data-Model-5_0_5.rtf

[7] Diessen, R. J. van (2002). 'Preservation requirements in a deposit system'
http://www.kb.nl/hrd/dd/dd_onderzoek/reports/3-preservation.pdf

[8] Hunter, J; Choudhurry, S. (2006). 'PANIC - An integrated approach to the preservation of composite digital objects using semantic web services', *International Journal on Digital Libraries,*Vol. 6 (2), pp.174–183.
http://www.itee.uq.edu.au/~eresearch/papers/2006/IJDL2006.pdf

[9] Franklin, B. (2006). 'Trial by File: Five Tools for Managing Formats', *RLG Diginews,* Vol.5 (10),
http://www.rlg.org/en/page.php?Page_ID=20987#article3

[10] The PA/4 working group (2007). 'Tool and service prototypes for internal evaluation', available on Planets Wiki at http://www.planets.arts.gla.ac.uk/private/pages/wiki/index.php/Image:PA4-D3.doc

[11] The PA/5 working group (2007). 'Starting point for registration and discovery of PA/5 tools', available on Planets Wiki at http://www.planets-project.eu/private/planets-ftp/WP_PA/PA5/PA5_startingpoints_registry_invocation_v0.2.doc