Project Number	IST-2006-033789
Project Title	PLANETS
Title of Deliverable	**Design for recommender system in Plato**
Deliverable Number	D6
Contributing Sub-project and Work-package	PP/6
Deliverable Dissemination Level	Internal PU
Deliverable Nature	Report
Contractual Delivery Date	31st June, 2009
Actual Delivery Date	30th September, 2008
Author(s)	Hannes Kulovits

Abstract

This document contains the first iteration for a design for recommender systems in the preservation planning tool Plato. The integration of such support systems shall reduce the effort needed to create a preservation plan and enable preservation planning for less experienced users.

.

Keyword list

Recommender Systems, Collaborative Filtering, Content-based Filtering, Preservation Planning, Plato

Contributors

Person	Role	Partner	Contribution
Hannes Kulovits	Main author	TUWIEN	Document owner and author
Rainer Simon	Reviewer	AIT	Reviewer for internal deliverable
Tim Gollins	WP Lead	TNA	Review and comments

Document Approval

Person	Role	Partner
Rainer Simon	Reviewer	AIT

Distribution

Person	Role	Partner
Planets Internal		

Revision History

Issue	Author	Date	Description
0.1	HK	31/08/2009	Version for internal review
0.2	HK	07/10/2009	Incorporated review comments

References

Ref.	Document	Date	Details and Version
1	Plato: A Service Oriented Decision Support System for Preservation Planning (2008).		
2	Aitken, B., Helwig, P., Jackson, A. N., Lindley, A., Nicchiarelli, E., and Ross, S. The planets testbed: Science for digital preservation. Code4Lib 1, 5 (June 2008). See http://journal.code4lib.org/articles/83.		
3	Balabanovic, M., and Shoham, Y. Combining content-based and collaborative recommendation. Communications of the ACM 40 (1997), 66–72.		
4	Becker, C., Rauber, A., Heydegger, V., Schnasse, J., and Thaller, M. A generic XML language for characterising objects to support digital preservation. In 23rd Annual ACM Symposium on Applied Computing (SAC'08) (Fortaleza, Brazil, March 16-20 2008).		
5	Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12, 4 (2002), 331–370.		
6	Farquhar, A., and Hockx-Yu, H. Planets: Integrated services for digital preservation. International Journal of Digital Curation 2, 2 (November 2007), 88–99.		
7	Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using collaborative filtering to weave an information tapestry. Commun. ACM 35, 12 (1992), 61–70.		
8	Hofman, H., Planets-PP subproject, Becker, C., Strodl, S., Kulovits, H., and Rauber, A. Preservation plan template. Tech. rep., The Planets project, 2008. http://www.ifs.tuwien.ac.at/dp/plato/docs/plan-template.pdf.		
9	ISO. Open archival information system – Reference model (ISO 14721:2003). International Standards Organization, 2003.		
10	Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., and Cohen, M. D. Intelligent information-sharing systems. Commun. ACM 30, 5		

	(1987), 390–402.		
11	Martin, D. Owl-s: Semantic markup for web services.		
12	Mirza, B. Jumping connections: A graph-theoretic model for recommender systems. Master's thesis, Computer Science Dept., Virginia Tech, 2001.		
13	Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: an open architecture for collaborative filtering of netnews. In CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work (New York, NY, USA, 1994), ACM, pp. 175–186.		
14	Resnick, P., and Varian, H. R. Recommender systems. Commun. ACM 40, 3 (1997), 56–58.		
15	Strodl, S., Becker, C., Neumayer, R., and Rauber, A. How to choose a digital preservation strategy: Evaluating a preservation planning procedure. In Proc. of JCDL'07 (June 2007), pp. 29–38.		
16	Thaller, M., Heydegger, V., Schnasse, J., Beyl, S., and Chudobkaite, E. Significant characteristics to abstract content: Long term preservation of information. In ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 41–49.		
17	Design for Automated Collection Profiling Service (2^{nd} Iteration). Planets deliverable PP6-D4, November 2008		
18	Report on the Conceptual Aspects of Preservation, based on Policy and Strategy Models for Libraries, Archives and Data Centres. Planets deliverable PP2-D3, July 2009.		
19	Perugini S., Gonalves M. A., and Fox E.A. (2004). Recommender Systems Research: A Connection-Centric Survey. Journal of Intelligent Information Systems, 23(2): 107–143.		
20	Terveen, L. and Hill, W. (2001). Beyond recommender systems: Helping people help each other. In Human-Computer Interaction in the New Millenium, J. Carroll, Ed. Addison-Wesley, Reading, MA, USA.		
21	Belkin, N., Croft, B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM 35(12) (1992) 29-38		
22	Alspector, J., Kolez, A., and Karunanithi, N. (1998). Comparing Feature-Based and Clique-Based User Models for Movie Selection. In Proceedings of the Third ACM Conference on Digital Libraries (pp. 11–18). Pittsburgh, PA: ACM Press.		
23	Mooney, R. and Roy, L. (2000). Content-Based Book Recommending Using Learning for Text Categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries (pp. 195–204). San Antonio, TX: ACM Press.		
24	Pazzani M., Muramatsu J., and Billsus D. (1996). Syskill & Webert: Identifying Interesting Web Sites. In Proceedings of the Thirteeth National Conference on Artifical Intelligence, pages 54–61. AAAI Press, 1996.		
25	Housman, E. and Kaskela, E. (1996). State of the Art in Selective Dissemination of Information. In Proceedings of the IEEE Transaction on Engineering and Writing Speech (pp. 100–112). Stanford, CA: AAAI Press.		
26	Pattenden-Fail J., Ballaux B., Balle Sørensen A., Kruse F., and Thøgersen J., Report on usage models for libraries, archives and data centres, results of the second iteration. Planets deliverable PP3-D2, August, 2008		
27	Kulovits H., Becker C., and Guttenbrunner M., Collection of Preservation Plans. Planets deliverable TB3-D8, April, 2009		

EXECUTIVE SUMMARY

Due to fast technological changes, digital objects face obsolescence quickly. This demands action from the person in charge to mitigate arising risks and ensure accessibility of the collection over the long term. To tackle this challenge different strategies such as migration and emulation have been proposed; however, the decision which one to adopt is complex and requires detailed knowledge and experience in digital preservation. The process of evaluating strategies against well-defined requirements is called preservation planning and shall support decision makers to opt for the right strategy. The result of this activity is a preservation plan which contains the taken decision including the complete evidence base. Even though tool support already exists, the creation of a preservation plan still is a complex and time-consuming endeavour due to demanding tasks such as finding potential preservation actions and eliciting the institutional requirements. This deliverable proposes conceptual enhancements in the planning process to integrate recommender systems in order to reduce the effort needed and enable preservation planning for users with less experience. Furthermore, the basis for the information filtering in the different areas is identified.

TABLE OF CONTENTS

1. Introduction

The creation of a preservation plan encompasses several essential activities which must be accomplished in order to arrive at a well-informed, consistent, comparable, and accountable recommendation for a preservation solution. Preservation planning is the process of evaluating potential solutions against specific requirements and building a plan for preserving a given set of objects. To date this is mostly done manually and in a rather unstructured way with little or no tool support. In the course of Planets a preservation planning methodology and a software tool (Plato) implementing this methodology are being developed to address these issues. The methodology supports the evaluation of preservation strategies and the production of well-documented, accountable recommendations on which strategy to follow.

The current version of Plato supports the planning workflow and integrates services for content identification and preservation action. The software itself is built upon the Planets Interoperability Framework that guarantees loose coupling of services and registries through standardised interfaces. Although the Planets preservation planning methodology is already well supported by Plato certain tasks are still very complex and require detailed knowledge from the planner. Tasks such as discovering potential preservation actions to consider for evaluation, and selecting representative sample objects to apply them on are very challenging.

This deliverable proposes integration blocks for recommender systems in Plato to further support the users in their preservation planning endeavours. The integration of recommender systems in Plato strives for four major goals:

- Enable preservation planning for inexperienced users. Building the objective tree which lies at the heart of the preservation plan demands considerable knowledge regarding digital preservation from the planner. The requirements must be measurable, general enough to not focus on a particular preservation strategy and specific enough to reflect the institution.
- Reduce complexity of certain workflow steps. The selection of preservation alternatives for instance can be quite complex as numerous candidates might be available. Not only tools wrapped as web services must be considered as alternatives but also migration paths, emulation view-paths and tools not yet wrapped as web service. Furthermore 'keep status quo' might also be considered as an alternative.
- Improve preservation action recommendation. As the quality of the recommended preservation action highly depends on the defined requirements to which the candidate solutions are evaluated against, particular attention should be paid to the process of requirements elicitation. In this stage a recommender component can advocate requirements that especially focus on the collection in question and are likely to be tautological regardless the institution.
- Reduce time and effort for planning activity. There is possibly a plethora of potential preservation actions available from which the planner has to choose for evaluation. The application of a recommender in this stage can support the planner by, for instance, filtering out non-applicable ones and selecting the top-N best performing services considering the institution's policy and requirements.

The remainder of this document is structured as follows. The next section outlines related work in the area of recommender systems. Section 3 gives an overview of the preservation planning workflow. Section 4 describes the integration of recommender systems in the planning workflow. Section 5 draws conclusions and points out directions for future work.

2. Related work

In systems where the number of choices can be enormous, recommender systems [14] assist users in identifying a subset of items from a typically larger set of possibilities they might be interested in. The main goal of recommenders is to reduce the complexity for individuals trying to find their way through large amounts of information and suggesting those pieces that were supposedly the most relevant ones. Furthermore, recommender systems take personalisation into account which is targeted on the user, as the need for information differs for each user. A computer scientist for instance, is, probably more interested in software engineering books than books on marketing and sales - contemplated from a professional point of view. A very common way to obtain recommendations is by word-of-mouth or by reading reviews about items one is interested in. Systems like amazon.com suggest products on that basis, depending on user profiles and recensions provided by known users.

Generally recommender systems are seen from four main dimensions [12]:

1. How the system is modelled, i.e. how the recommendations are made.

2. How a recommender system is targeted, i.e. the level at which information is tailored.

3. How a recommender system is built.

4. How a recommender system is maintained (online vs. offline)

A commonly accepted classification distinguishes between content-based and collaborative filtering. The former, which is also known as cognitive filtering, is deeply rooted in information retrieval. It calculates similarities between a number of items a user appreciates, and the products that are not yet known to the user. The latter connects groups of users with similar preferences or interests to take advantage of the group's experiences. In contrast to content based filtering, collaborative filtering calculates similarities between the users based on the available user profile. A recommender system can also follow a combination of two or more different approaches (hybrid recommender) under a single framework in order to leverage the advantages of the individual one and address the disadvantages of each.

2.1 Collaborative filtering

The phrase collaborative filtering has been coined by Goldberg et al. while describing Tapestry [7], an experimental mail system which they developed at the Xerox Palo Alto Research Center. Tapestry later became known as the first recommender system [14]. The system performs filtering by considering users' reactions, called annotations, to documents they read. GroupLens (Konstan et al., 1997), which is another historically important system, helps readers of Usenet newsgroups find articles they will probably like amongst a huge amount of available articles. It facilitates the rating of articles by users after they read them and computes correlation between readers by comparing these ratings. Collaborative filtering is also called social filtering [13].

One significant difference, compared to the content-based approach is that no representation of the items in term of features is needed. A collaborative filtering recommender (Figure 1) can be seen as a process that takes a user representation and a generic set of artefacts as input and transforms this to a recommendation of a subset of these artefacts [19].

Figure 1 Collaborative filtering

According to Terveen and Hill [20] three requirements have to be met to support collaborative filtering:

1. Many people must participate in order to make it possible that any given person will find other people with similar preferences.
2. The system must enable an easy way for users to represent their interests.
3. Algorithms applied in the system must be capable of matching people with similar interests.

2.2 Content-based approach

The very first content-based recommender systems were mostly based on information retrieval techniques such as `tf*idf`. Content-based filtering is also known as cognitive filtering [10] and calculates similarities between a number of items a user appreciates, and the products that are not yet known to the user. They are referred to as performing *information filtering* [21] and have later been known to the recommender systems community as content-based filtering. The basis of recommendations given by a content-based recommender system is formed by two things:

1. A definition of the features associated with the object.
2. A profile of the user's interest. This profile is based upon the features present in objects rated by the user.

Such systems have been applied in many contexts, from recommending movies [22] and books [23] to recommending web sites [24]. One of the first forms of information filtering systems was based on *keyword matching* and was called *Selective Dissemination of Information* (SDI) [25]. The system was designed to support researchers in spotting current publications by alerting the user when a new document was published in a specific field of research. After creating a user account and feeding the system with a set of keywords that best match the user's field of interest the system periodically informs the user when a new publication matches her profile.

2.3 Hybrid Recommender Systems

Hybrid recommender systems combine two or more recommendation approaches under a single framework in order to leverage the advantages of the individual one and address the disadvantages of each. Burke [5] surveys numerous ways of combining different approaches whereas the most popular amongst them is a composite of *collaboration* and *content*. One of the first hybrid recommenders is Fab [3] with the aim of proposing web sites to its users.

Amazon for instance applies a hybrid recommender which combines content-based and collaborative techniques. When coming across the "Gang of Four's" book *"Design Patterns: Elements of Reusable Object-Oriented Software"* while exploring Amazon's range, further more books appear as recommendation: "Users who bought *Design Patterns: ...* also bought ..." The system behind this recommendation calculates similarities between items bought by customers in the past as well as similarities for rated items. As these calculations are computationally intensive they are constantly performed in the background (*offline*) and collected in a knowledge database. When a user takes a look at a specific item the recommender system looks up the knowledge base for similar items and presents those to the user. Such a combination is capable of giving meaningful recommendations in very short time.

3. Preservation Planning and Plato

The systematic planning approach developed within the Planets project aims for evaluating potential alternatives for preservation actions and building thoroughly-defined, accountable preservation plans for keeping digital content alive over time. In this approach, preservation planners empirically evaluate potential action components in a controlled setting and select the most suitable one with respect to the particular requirements of their institution [15]. The procedure is independent of the solutions considered; it can be applied for any class of strategy, be it migration or emulation or different approaches and follows a variation of utility analysis. The selection procedure leads to well-documented, well-argued and transparent decisions that can be reproduced and revisited at a later point in time.

The planning tool Plato [1] implements the preservation planning workflow and supports, documents, and automates the decision procedure. Following the planning process in Plato results in a well-documented preservation plan [8] one can be held accountable for. The software itself has been implemented as a JavaEE compliant web application relying on open frameworks such as Java Server Faces and AJAX for the presentation layer and Enterprise Java Beans for the backend. It is integrated in the Planets interoperability framework that supports loose coupling of services and registries through standard interfaces and provides common services such as user management, security, and a common workspace. Based on this technical foundation, the aim is to create an interactive and highly supportive software environment that advances the insight of preservation planners and enables proactive preservation planning.

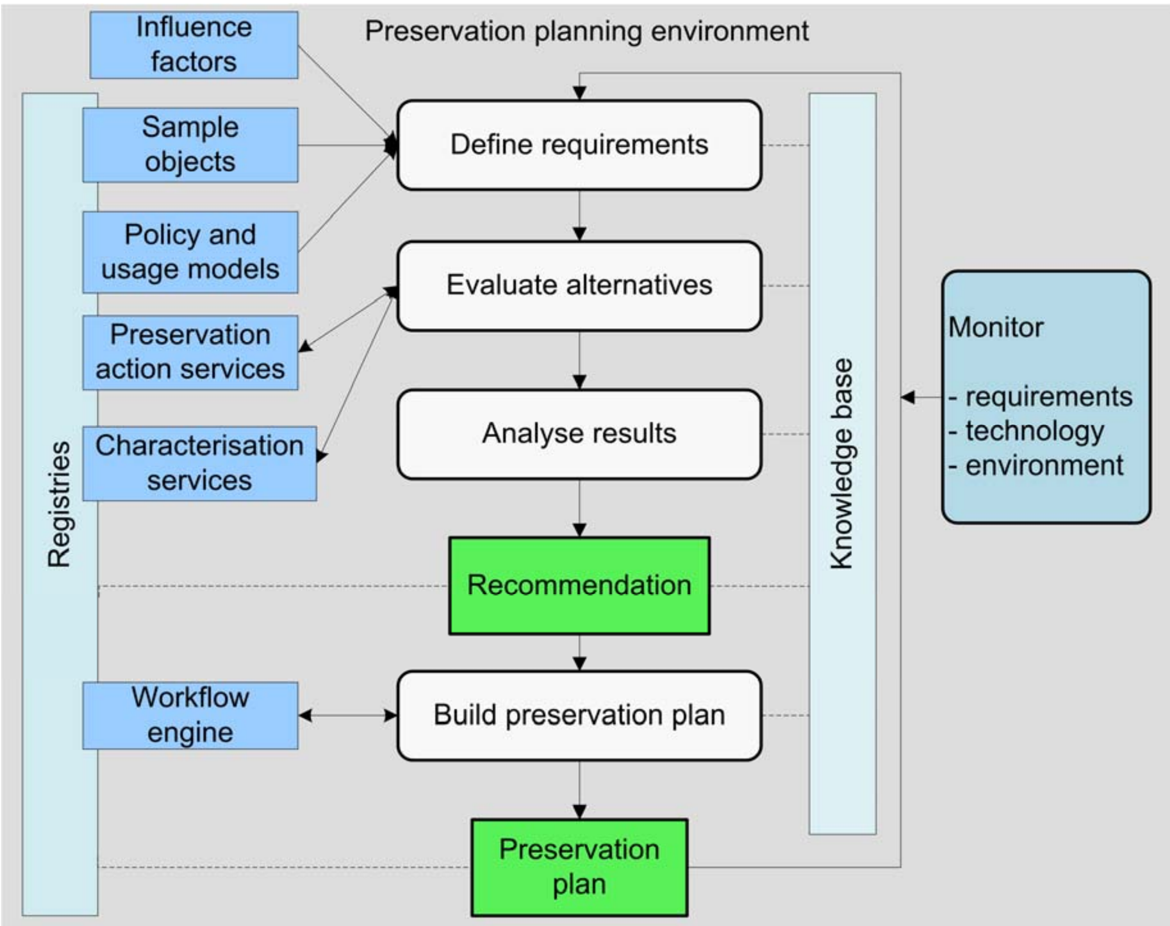

Figure 2 Preservation planning workflow

Figure 2 shows the preservation planning environment with the workflow and the relevant entities and repositories influencing the respective phases. The planning workflow that leads to the preservation plan prescribes four phases:

1. Define requirements. The first phase documents constraints and influence factors on potential preservation strategies. It then continues with a thorough description of the collection and the chosen sample objects from that collection and concludes with the definition of the complete set of requirements. At the end of this phase the planner has a detailed and exact understanding of the collection and the preservation goals. The elicitation of the institution's requirements is the core activity in the planning workflow and vital as the requirements co-determine the optimal preservation action within the institution's context. Not all university libraries or national libraries for instance share the same objectives.
2. Evaluate alternatives. The second phase starts with discovering potential preservation actions (alternatives) which are then evaluated in a quantitative way. Controlled experiments are carried out, applying the alternatives to the defined sample objects and analysing the outcomes with respect to the requirements. The result of this phase is an evidence base that underlies all decisions to be taken in the subsequent phases.
3. Analyse results. In the third phase the results of the experiments are analysed and aggregated. The result of this phase is a ranked list of alternatives whereas that alternative with the highest performance value presents the recommended preservation action.
4. Build preservation plan. In the final phase, based on the recommended preservation action a preservation plan is created which corresponds to the Develop Packaging Designs and Migration Plans functionality in the OAIS model [9].

Another important role in the planning process play registries holding web services for preservation action and characterisation that can be run on digital objects. While characterisation services such as DROID[1] (also described in Planets deliverable PP6-D4 [17]) and JHove are used to understand the digital objects on hand, action services are being evaluated and selected based on characterisation results. Furthermore the planning is supported by a knowledge base that holds reusable patterns and templates for requirements recurring in different planning situations.

4. Recommender Systems in Plato

4.1 Introduction

The process of creating a preservation plan in Plato can benefit from recommender systems in 5 different areas:

1. Recommend particular tree template and/or fragments.
2. Reduce the total amount of preservation action candidates by recommending top-N preservation action services to the user.
3. Recommend representative sample objects from a collection of objects.
4. Recommend properties that can automatically be measured to be mapped to user requirements.
5. Recommend transformation templates.

The recommendations given by the system are based on different sources of information available to the recommender component. Five different sources of information have been identified so far that can be used as a basis for deriving a recommendation:

[1] http://droid.sourceforge.net

- *Usage model.* Describes in a machine interpretable way how the different users work with the collection and which priorities they have. Figure 3 shows the current version of the usage model which is implemented in Plato. It is described in more detail in the Planets deliverable PP3-D2 [26].

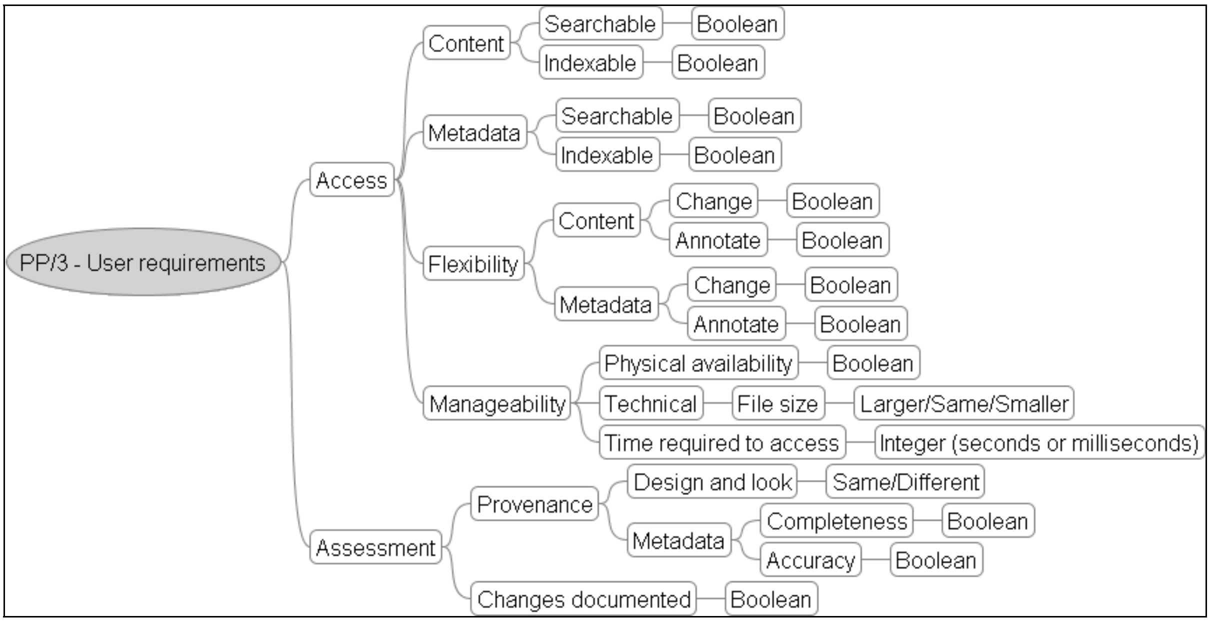

Figure 3 Usage model

- *Collection profile.* Describes the characteristics of the objects in the collection and the distribution of object types within the collection.
- *Policy model.* Captures the overall organisational characteristics and requirements of the repository. The current implementation of the policy model in Plato is a pure static one and so far serves as a documentation of the institution's policies. The policy model developed within Planets is described in more detail in deliverable PP2-D3 'Report on the Conceptual Aspects of Preservation, Based on Policy and Strategy Models for Libraries, Archives and Data Centres" [18].
- *Knowledge base.* The knowledge database provides tree templates and fragments as well as templates for transformation table and potential preservation alternatives. The database stores the gained experience and knowledge from earlier preservation planning activities.
- *Testbed Results.* Benchmark data that allow an objective ranking of preservation action services.

The recommender component in Plato follows a hybrid approach building upon the different users and institutions working with the central instance of the planning tool as well as the various features of the items to be recommended (requirements, preservation action services, etc.). Following the 5 different areas of integration are explained in more detail.

4.2 Recommend tree template and/or fragments

The objective tree, which is the basis of the preservation planning workflow, is usually created in a workshop setting with stakeholders from different domains. All of them contribute to the requirements gathering process. The elaborated requirements tree documents the individual preservation requirements of an institution for a given homogeneous collection of objects. [15] report on a series of case studies and describe objective trees created in these.

In the stage of eliciting the requirements the preservation planner can benefit from recommendations concerning the requirements tree either in the form of templates or tree fragments. Template trees represent best practice branches and sub-trees of specific planning contexts such as for different institutions or different types of digital objects. These templates can be used as a starting point to build new objective trees as well as for refinement of existing objective trees. The templates can be adjusted for each respective preservation context. Planets deliverable TB3-D8 [27] reports on further preservation planning case studies which were all feed into the Plato knowledge base as tree templates and fragments. The knowledge base as an integral part of the planning environment can be seen in Figure 2. The decision which template/fragment to recommend is influenced by four entities:

- *Policy model.* Certain preservation policies can be adopted as requirement. A preservation policy which specifies the preservation strategy to follow (e.g. migration or emulation) can be translated into a requirement and associated with a measurement scale. The same is true for policies determining an open-source strategy which can be adopted as a requirement for applied preservation actions.
- *Usage model.* The recommended requirements can be further extended including knowledge about how the various users work with the collection.
- *Chosen sample objects.* The description of the collection the preservation plan is created for and the chosen sample objects from that collection determine the content family: application, audio, video, image, or text. Each content family contains a representative pre-defined template tree that can be suggested to the planner.

4.3 Recommend Representative Sample Objects from a Collection

In the second step of the preservation planning workflow ('Define Sample Records') the planner selects sample records representing the variety of object characteristics of the considered collection. These samples are later used for evaluating the preservation alternatives. As the experiments and the evaluation of the outcome depend on the selected sample records they have to be chosen advisedly. A comprehensive collection profile using DROID for identification and FITS[2] for metadata extraction will be created. Based on that profile the system selects a minimal set of sample objects covering a maximum number of object characteristics.

4.4 Recommend Top-N preservation action services

Discovering potential preservation actions is one of the most challenging and time-consuming tasks in the planning process. Numerous tools are available that come into question, each of them need specific input parameters and rely on a particular environment. To find relevant preservation action services the planner has to bear all these constraints in mind and rifle through existing preservation action registries. A recommender system in this stage of the workflow can reduce the amount of potential preservation action services available to the planner by recommending top-N preservation action services. The recommendation will be a ranking that is based on:

- *Migration path.* Direct migration with no intermediate conversion are preferred and thus ranked higher.
- *Collection profile.* The sample objects and the definition of the collection the preservation plan is created for determine the file format the action service must be able to handle; others can be filtered out.
- *Requirements defined by the planner and institutional policies.* For example, based on an institutional policy advising that only migration strategy shall be applied emulation services can be filtered out. This reduces the amount of candidate preservation actions and alleviates the decision which alternatives to choose for evaluation.
- *Test results.* Objective evaluation results produced by experiments carried out in the Planets Testbed[6, 2] are considered in the ranking of potential preservation actions. Actions which performed better in an experiment on comparable objects are ranked higher.

4.5 Recommend mapping

Comparison services such as the comparator developed in the course of the eXtensible Characterisation Language [4, 16] specify measurable properties as well as property-specific

[2] `http://code.google.com/p/fits/`

metrics and their implementation as algorithms in order to identify degrees of equality between two objects. This is in principle independent of the applied strategy, i.e. migration or emulation. The compared objects can be both the original and a migrated object, or the original object in two different environments for emulation. To allow comparison and evaluation, a mapping is created between the requirements specified in the objective tree and the characteristics that can be measured and compared automatically by the available characterisation tools. At present this is done manually for each leaf criterion. Each leaf criterion can be mapped to a property that can be measured automatically.

The mapping done in this workflow step can be supported by a recommender system in some very basic way using pattern matching. Two possible example scenarios are given below:

- An automatically measurable property named `imageHeight` can be recommended as a candidate property to be mapped to requirements named height of image, image's height or height when it appears in a sub tree called object characteristics.
- A requirement named `Size` in a sub-tree object characteristics can be suggested to be split up into `imageHeight` and `imageWidth` because those can be measured automatically by a comparator service.

4.6 Transformation templates

Requirements are measured in different scales and are made comparable by mapping to a uniform scale using transformation tables. The resulting scale might for instance range from 0 to 5. A value of 0 denotes an unacceptable result and thus serves as a drop-out criterion for the whole preservation alternative. This transformation has to be done for each leaf criterion in the objective tree.

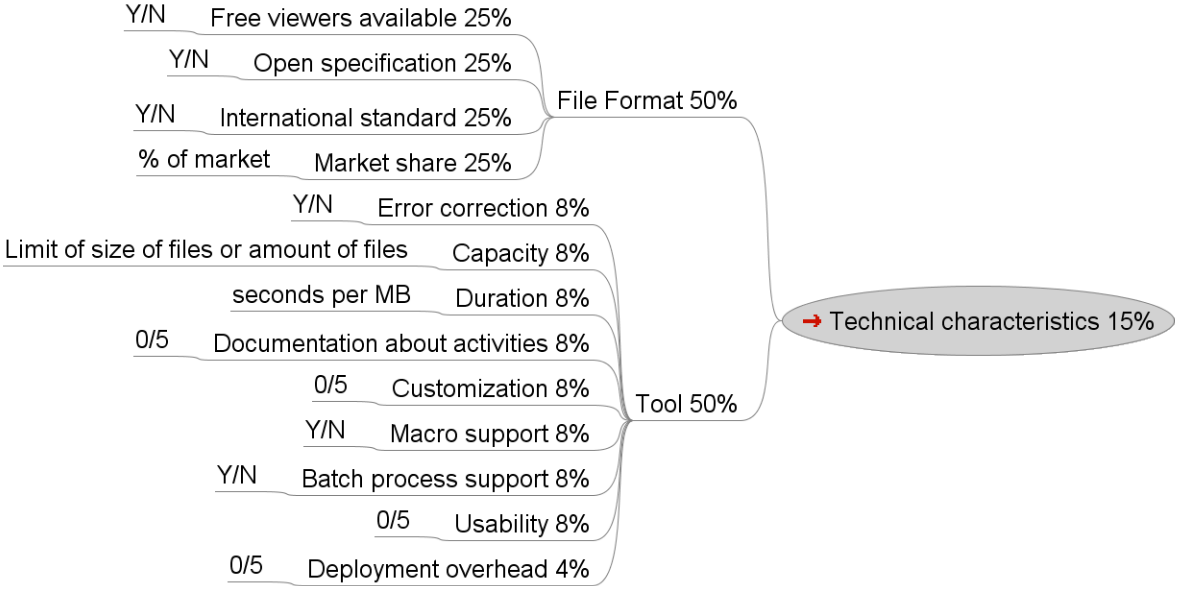

Figure 4 Technical characteristics in objective tree

Figure 4 illustrates a branch, focussing on technical characteristics of the collection, which has been taken from a requirements tree of a specific institution. In a requirements tree the leaves of the tree determine the scale of the respective requirement, ranging from Y/N (either yes or no), percent of market share, to seconds per MB. The requirement 'Open specification' refers to the openness of the file format specification which can either be 'Yes' (the specification is publicly available) or 'No' (the specification is proprietary). Two different institutions can have two different transformation tables for the same requirements. One institution for instance can define a proprietary format specification as a drop-out criterion and another one as still acceptable. The first institution would map 'No' to zero and the second one to a value larger than zero, e.g. 1.0. The system when recommending a transformation table must consider:

- *Policy model.* The institutional policies allow the system to differentiate between acceptable and not acceptable criterions.

- *Usage model.* Similar to the policies the usage profile defines if a requirement will be defined as a drop-out criterion. The usage profile for instance declares that users perform full text search. Loosing searchability by a preservation action shall thus be not acceptable.
- *Transformation tables* established by other institutions stored in the knowledge base. The same or slightly adapted transformation tables can often be adapted for similar institutions.

5. Summary

This deliverables outlines specific integration blocks for recommender systems in the preservation planning tool Plato. On the one hand by this integration the effort needed to create a thorough preservation plan can be reduced. On the other hand it furthermore enables preservation planning for users with less experience in digital preservation. The next steps towards recommendation supported preservation planning are:

1. The current implementation of preservation policies in Plato is based on an extensive tree describing the policies in a structured but static way. Import from the openly available Freemind tool is possible. In that tree a policy consists of the policy statement and a freely defined measurement scale to which degree the policy applies. However, to avoid ambiguity policies that potentially influence decisions made during the planning process have to be integrated into the system and associated with a unique identifier. This allows the recommender to use those policies for filtering results.
2. A detailed collection profile and risk analysis is vital for the planning process because the selection of potential preservation actions and the experimentation depend on it. Existing characterisation tools such as DROID and FITS will be used to extract the relevant object characteristics and create a collection profile that can then be used by the recommender.
3. Existing registries hold web service descriptions of preservation actions. Besides the technical information a WSDL contains, additional information it optionally carries, about supported data types and operations, is mainly in natural language. To enable an automated selection of preservation action services they must be enriched with QoS information and metadata such as licensing and required platform of the underlying tool. OWL-S [11], an ontology adopted for web services, may be used to create a computer-interpretable description of these web services to allow proper recommendations.