Modeling and Building Personalized Digital Libraries with PIPE and 5SL

Marcos André Gonçalves, Ali A. Zafer, Naren Ramakrishnan, and Edward A. Fox[1]

Abstract: We present an integrated personalization framework which enables the building of personalized digital libraries (DLs) targeted for different communities of users. This framework relies on two major building blocks: 1) PIPE, a customizable methodology for personalization that supports the construction of automatic personalized views of a DL without enumerating explicit restructurings or interaction sequences; and 2) 5SL, a declarative language with a formal theoretical basis, which allows specification and semi-automatic generation of digital library applications. We analyze personalization opportunities in the DL context according to our formal theory and describe how the proposed framework can explore those opportunities. Some initial case studies and experiments that show the feasibility of our proposal also are reported.

1. Introduction

Digital libraries (DLs) are enormous information warehouses, with huge amounts of data, encompassing many kinds of multimedia formats. Societies of users/patrons are inundated with massive quantities of information and are rarely provided with effective tools that allow them to customize DL services and content for their own interests. Personalization [1, 2], a possible approach to the problem, involves techniques and mechanisms to reduce this information overload and tailor DL systems for particular user communities with specific interests. Nevertheless, DLs are extremely complex information systems; they integrate findings from disciplines such as hypertext, information retrieval, multimedia services, database management, and human-computer interaction. There are several current DL systems [3-5], most of them with inflexible architectures and poor interoperability. This is due mainly to the lack of sophisticated theories and models for the field. Incorporating personalization as a basic DL capability thus becomes a Herculean task. Exacerbating the problem, personalization similarly suffers from the lack of models and methodologies to guide DL designers desiring to integrate it in their projects.

In this paper, we propose a solution for DL personalization. The solution involves an integrated framework that enables the automatic building of personalized digital libraries targeted for different communities of users. This framework relies on two major building blocks: 1) PIPE [6], a customizable methodology for personalization that supports the building of automatic personalized views of the DL without enumerating explicit restructurings or interaction sequences; and 2) 5SL, a declarative language for specification and generation of digital library applications. 5SL is based on the 5S formal theory of Streams, Structures, Spaces, Scenarios and Societies [7], which provides a unified view of the DL field. A common theme in 5SL and PIPE is the factorization of an information space in terms of its structural and functional interaction metaphors. We explain how to combine these complementary approaches and how to manipulate resulting decompositions in effective ways.

This paper is organized as follows. In section 2, we present the 5S theory and use it to analyze and understand possible applications of personalization in a DL environment. In section 3, we describe PIPE, 5SL and how we combine them in our proposed integrated personalization framework. In section 4, we illustrate the feasibility of our framework through some initial case studies. Section 5 explores future work.

2. Understanding DL personalization through 5S glasses

Here, we briefly and informally introduce the 5S model of Streams, Structures, Spaces, Scenarios, and Societies. 5S aims to define digital libraries rigorously and usefully and provides a practical unification for the field. Streams in 5S are sequences of abstract items, involving static and dynamic content and are useful for text, multimedia information, etc. Structures can be defined as labeled directed graphs, which impose organization. Spaces define sets of arbitrary objects and operations on those sets that obey certain rules. Scenarios consist of sequences of events or actions that modify states of a computation in order to accomplish a functional requirement, therefore defining DL services. Societies involve entities (such as individuals or automated agents) and the relationships between and among them.

This model allows us to better understand DL issues and also provides a set of abstractions to build more complex DL concepts. For example, in [7] we formally define DL concepts such as digital objects, structural

[1] Virginia Tech, Department of Computer Science, Blacksburg, VA 24061, USA

and descriptive metadata, repositories, collections and services in terms of the 5S abstractions. 5S fulfills the role of an analytical tool useful for understanding personalization aspects in the DL context. Thus, first analyzing the low-level of personalization that can occur over the basic 'Ss', possible personalization activities would include:

1) Personalization of streams: which could include, in the case of textual streams, translations of language and conversion of encodings, or, in the case of multimedia data, possible conversions between formats according to a user's platform;
2) Personalization of structures: including restructuring, reduction, or other transformations over classification systems, ontologies, internal structures of documents, etc;
3) Personalization on spaces: such as mappings between different spaces (e.g., from vector space models to probabilistic ones) for interoperability or reduction of dimensionality for providing better search services (e.g., with Latent Semantic Indexing (LSI) [8]);
4) Personalization on scenarios: like scenario re-design, by introducing new functions and interaction techniques, e.g., navigation by context [9], or by specializing existing ones, e.g., changes in syntax and parameters for searching [10];
5) Personalization on societies, where all other personalization dimensions would be organized or targeted for particular societies of users, e.g., incorporation and adaptation of specialized services for librarians, professors, and students in a digital library of theses and dissertations.

Even more interesting is the rich personalization environment that can be built over the compositions of those basic personalization operations when considering more complex DL concepts that involve multiple "Ss". For example, the abstraction of a *digital object* as a composition of different structures imposed over several streams of data and associated metadata, reveals a number of opportunities for personalization, e.g., different dissemination services/scenarios that transform internal structures and streams in order to provide diverse customized presentations.

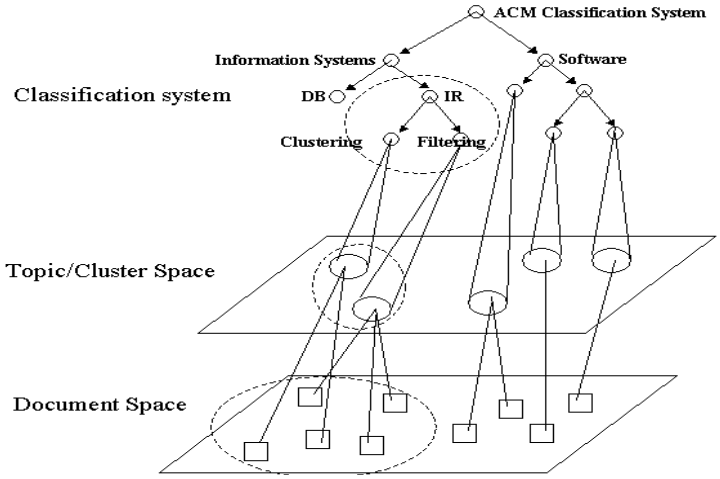

Figure 1. Personalization of a classification system

Another example includes combination of structures, like hierarchical classification systems, with filtering/searching services and space mappings to provide solutions for information overload. An example of this is seen in Figure 1, which shows a simplified portion of the ACM Classification System connected to a cluster space and indirectly to a document space. Some input from the user (e.g., statement of interest in Information Retrieval (IR)) could allow the personalization mechanism to isolate the user interests in just a small part of the whole classification system (the portion circulated with shaded lines). Linking between the entries of the classification system, the topic/cluster space, and the document space leads to a halving of the browsing space.

3. The Integrated Personalization Framework

The idea here is to provide automatic personalized views of the DL targeted to particular societies of users. The approach described can be related to previous works on automatic generation of web sites based on views [11-13] or automatic customization of user interfaces based on parameterized views [14]. The differences here are: 1) we utilize a novel, theoretical approach for declarative specification of digital library applications; and 2) we incorporate an original customizable methodology for automatic personalization of DLs based on user input, which does not require enumerating explicit restructurings or interaction sequences beforehand. In other words,

the DL designer is not required to anticipate and predefine mechanisms for every conceivable user partial input. For example, in the case of parameterized views, the designer is responsible for explicitly enumerate the parameters that are associated with a particular view; the actual parameter values are evaluated when the view is computed, retrieving all database objects that satisfy the query with respect to those values. Contrarily, our methodology provides richer personalization capabilities once it is able to factor an information space in terms of its structural and functional interaction metaphors and apply several different transformations over those decompositions. In the following, we describe the two main ideas behind our solution and how we combine them in our proposed integrated personalization framework.

3.1 The PIPE Methodology

PIPE provides a systematic conceptual methodology to study the design, implementation, and evaluation of personalization systems. PIPE models personalization by the programmatic notion of *partial evaluation* [15]. Partial evaluation is a technique used to automatically specialize programs, given incomplete information about their input. The input to a partial evaluator is a program and some static information about its arguments. The output is a specialized version of the program (typically in the same language) that uses the static information to "precompile" as many operations as possible. A simple example is how the C function `pow` can be specialized to create a new function, say pow2, that computes the square of an integer. Consider for example, the definition of a power function shown in the left part of Fig. 2(grossly simplified for presentation purposes). If we knew that a particular user will utilize it only for computing squares of integers, we could specialize it (for that user) to produce the pow2 function. Thus, pow2 is obtained automatically (not by a human programmer) from pow by precomputing all expressions that involve exponent, unfolding the for-loop, and by various other compiler transformations such as copy propagation and forward substitution. Partial evaluation is traditionally used to speed up a program and/or remove interpretation overhead, but it can also be viewed as a technique to simplify program presentation by removing information that doesn't apply to a particular user or is otherwise unnecessary.

```
int pow(int base, int exponent) {
    int prod=1;
    for (int i=0; i < exponent; i++)
          prod = prod * base;
    return (prod)}
```

```
int pow2(int base) {
     return (base*base);
}
```

Figure 2: Illustration of the partial evaluation technique. A general purpose **pow**er function written in C (left) and its specialized version (with `exponent` = 2) to handle squares (right). Such specializations are performed automatically by partial evaluators such as C-Mix.

Figure 3 shows another example of the approach with respect to the example considered in Fig. 1. The mutually exclusive dichotomies of the nodes representing topics in the classification system are modeled by `if else` statements in the corresponding programmatic representation. In the leaves, query functions or sets of handles of digital objects, which, respectively, implicitly and explicitly define the topic/group, could be stored. An example of the latter is the `doc.subject().match()` method, where `subject()` returns a subject object that has its own match method. The input of a user, corresponding to a variable associated with the desired topic or a related concept allows the personalization of the structure, via partial evaluation of the corresponding program. Some actual examples of this process are discussed in Section 4.

```
if(Information_Systems)
 {
  if(DB)
   …
  else if (IR)
    {
     if (Clustering)
       docs.subject().match("Clustering");
     else if (Filtering)
       docs.subject().match("Filtering");
    }
 }
else if(Software)
 {  …
 }
```

```
if (clustering)
   docs.subject().match("Clustering");
else if (Filtering)
   docs.subject().match("Filtering");
```

Figure 3. Corresponding programmatic representation of the classification system of Figure 1and output (left) of the partial evaluator (with subject = "IR")

3.2 5SL

5SL is a declarative language with a formal semantics for specification and construction of digital libraries. The semantics are understood in terms of a mapping of language constructs into the 5S framework. Its formal basis provides an unambiguous and precise DL specification tool, which can facilitate prototyping, allowing proofs of assertions and validation of implementations.

DL specifications, in 5SL, can be fed into our DL generator (see Figure 4(a)), to produce tailored DLs, suitable for specific platforms and requirements. These are built upon a collection of stock parts and configurable objects that provide the infrastructure for the new DL. This infrastructure includes the classes of objects and relationships that make up the DL. Also included are processing tools to create the actual library collection from raw documents as well as services for search, browsing, and collection maintenance.

We use 5SL for building XML-based digital libraries with advanced IR services. The framework shown in Figure 3 also supports semi-automatic generation of wrappers for harvesting environments. In one of those applications, a complete 5SL description was formulated for sites involved in the Networked Digital Library of Theses and Dissertations (see also Section 4.2). A specific DL generator targeted to the MARIAN Digital Library system [3, 16, 17] also was developed.

3.3 Integrated Personalization Architecture

The key idea here is to explore the commonalities of both strategies - they both are based on a *modeling* and a *factoring* of their respective problem domains - and to combine their complementary modes of exploring and manipulating the resulting decompositions in effective ways. 5S decomposes a DL in its most basic components, which are used as building blocks in 5SL to construct more complex DL components. The corresponding PIPE's factoring of the information infrastructure into programmatic constructs allows simple mappings of representations to connect both frameworks to leverage their respective benefits.

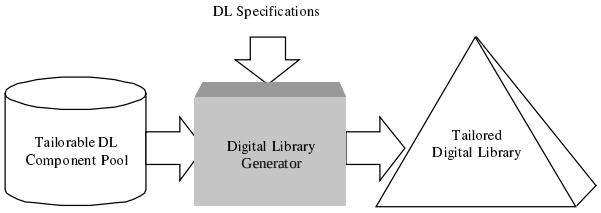

Figure 4(a). Digital library generation with 5SL

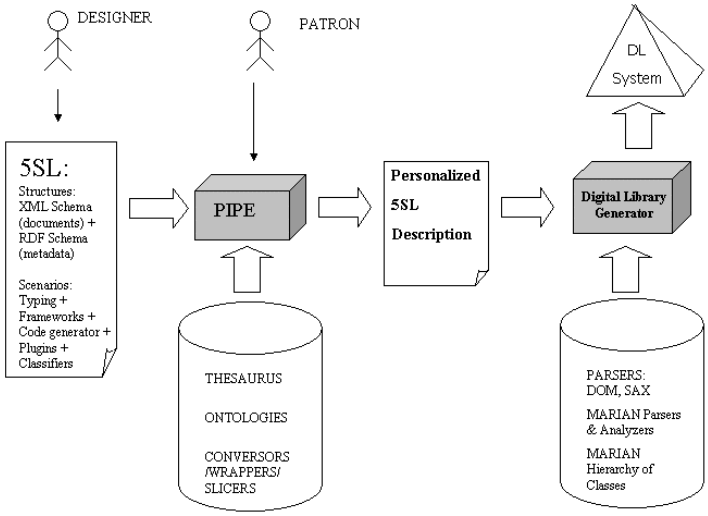

Figure 4(b).The Integrated DL Personalization Framework

Our Integrated Personalization Architecture is shown in Figure 4. It is an extension of the original architecture of Figure 4(b). 5SL encodes the original description of the DL as provided by the DL designer. In order to personalize it with PIPE, 5SL descriptions for structures and scenarios are modeled as programs, which abstracts the underlying schema and flow of information in the DL. Here we take advantage of the more stable, structured and managed nature of the DL and of our language. PIPE has originally been used to personalize Web sites and a great amount of effort was put on mining of semi-structured data [18]. In our case the structure of the DL is already explicitly captured by the 5SL description. Moreover, because 5SL uses standard XML syntax, we can take advantage of the architecture and reuse components of the component pool like standard parsers and wrappers, just redirecting their output, therefore enormously facilitating our task. The extended PIPE can also incorporate additional utensils like thesaurus and ontologies that would help to close the gap due to the vocabulary mismatch between the user's mental model and 5SL descriptions, for example, for labels used in structures defining classification systems and document organization. The 5SL programs are partially evaluated with respect to user input and the output is used to recreate a personalized 5SL description from the specialized program. This personalized description is then utilized to create a personalized DL matching user interests.

4. Case Studies

Our first experiments with the proposed framework explored the dichotomy between scenarios and structures: we personalize simple browsing activities performed over hierarchical classification systems. The use of classified subject lists in digital libraries, in greater or lesser extension, is almost ubiquitous and provides a number of advantages and obvious opportunities for exploration with our framework, e.g., 1) classification systems provide a taxonomic basis for the universe of the discourse, helping to better characterize it, and furnishing a natural and systematic factoring/partitioning of the information space; 2) as a consequence, they provide a natural browsing metaphor, thus offering systematic organization and presentation of resources that allows users to retrieve relevant resources more quickly and easily; and 3) it provides a description of the domain from which it is possible to characterize accurately the need for (and the role of) personalization.

4.1 Case 1: Library of Congress Classification System & Virginia Tech Library Catalog

Our first case study tried to prove the feasibility of our entire framework in a controlled setting. For this, we used the Library of Congress Classification Hierarchy (LCC) as the structure to be personalized and over which browsing scenarios would take place. The LCC was originally designed as a controlled vocabulary for representing the subject and form of the books and serials in the Library of Congress collection. The electronic version of the scheme is represented in MARC21 format and contains more than 450000 records for different categories or topics and hierarchical paths with length greater than 30 levels. In the case study, the classification system was to be used as a browsing tool to navigate across the On-Line Catalog of the Virginia Tech Library stored as MARC records in the MARIAN Digital Library system which has not previously offered a browsing option for this collection. The advantage here was that every entry in that particular catalog has been previously classified using the LCC, therefore providing a directed connection between the external classification scheme, the collection and the DL system (Figure 1). For the experimental setup, we first mapped a formerly available electronic version of the LCC, from MARC21 Format to XML resulting in a 200 megabytes file, which could be represented in 5SL and would allow the use of our XML parsers and validators from the component pool. For this task, we used the freeware MARCtoXML software developed by the Digital Library Research Laboratory at Virginia Tech and OCLC. After that, the event-driven SAX XML parser was driven to produce C code from the XML file in order to be partially evaluated as in the example in section 3.1. Each node in the resulting program (`else if` entry) was then associated to a MARIAN query of the form "`subject=node_label`", such that when clicked in the final HTML representation the corresponding query was sent to MARIAN. The DL generator materializes the personalized classification scheme by rendering it to HTML pages produced by associating specific Cascade Style Sheets (CSS) with the XML results. We also took advantage of the complete availability of classification scheme and have used "see also" and other relationships to help with the problem of vocabulary mismatch. So far, we were able to successfully personalize the classification system structures with entries of the user and to connect the personalized result with the underlying collection and DL system. However scalability was realized to be a serious problem.

4.2 Case 2: ACM Classification System and Networked Digital Library of Theses and Dissertations (NDLTD)

The objective of this case study was to test the same kind of application using our framework in a more uncontrolled setting. Hence, we used the ACM classification system with a subset of NDLTD collection, which contained just theses and dissertations about computer science and engineering, again stored in the MARIAN DL system. The difficulty here relied in the fact that theses/dissertations in this collection have not been classified with any controlled vocabulary, since the information about the work was entered by the author (student) herself as a way of gaining familiarity with digital library archiving concepts. Therefore, there was not any directed link between entries in the ACM scheme, topics and documents. With respect to Figure 1, it means that the links between the spaces and the classification system structure could not be directly and easily defined. The main idea here was to consider each topic of the classification system as a potential query to send to MARIAN. This approach proved to be too naive. To solve the problem, we utilized a combination of different strategies. In order to improve recall we expanded the topic query with terms from a *similarity thesaurus* [19]. The thesaurus represents term similarities, which reflect domain knowledge of the collection over which the thesaurus and is constructed based on *how terms are indexed by the documents*, i.e., the role of terms and documents are exchanged regarding the traditional view of IR. We used the similarity thesaurus to expand the query with the terms that are more related to the topic concept (computed as the centroid of the respective representing vectors of each term in the topic as expressed in the thesaurus). To improve precision, we made use of the powerful query mechanism of MARIAN, which can represent structured queries with weighted directed graphs. Therefore each topic was represented in the corresponding C program by an expanded graph query where weights associated with links captured the semantics and strength behind the fact that different parts of the document contribute differently to attest the main document topics. For example, terms identified from the title of a

document are more descriptive than terms identified from its abstract and less than its keywords or subject section. Another useful side effect of the building of the similarity thesaurus is that its entries could be used in the reverse way to help with the vocabulary mismatch problem. As in the previous case study, PIPE was successfully able to personalize DL structures, but the connection of the personalized results with the DL document space proved to be a difficult problem, requiring intricate solutions and relying in a great extension on the searching capabilities of the particular DL system. Also, the size of the sample collection was too small to consider any qualitative or quantitative measures, despite empirically verified improvements in results. Further studies for similar cases with bigger collections and different DL systems with varying capabilities are necessary before we can generalize the presented solution or provide guidelines for similar cases.

5. Future Work

As described in the analysis of section 2, there is a rich environment and a number of personalization opportunities in the DL setting. We intend to continue exploring many of those opportunities with our integrated personalization framework to completely evaluate its possibilities and limitations. For example, we will explore personalization of internal structures of documents and metadata, refinement of scenarios, etc. Particularly, in the case of personalization of classification systems, we intend to explore and combine additional techniques like LSI, feature selection [20] and automatic classification with discriminant analysis [21] to improve our solutions for vocabulary and ontological mismatches, query expansion and automatic classification. We intend to explore other programmatic techniques like *slicing* [22] to provide bottom-up inverse personalization capabilities, like for example to provide extensional inference capabilities to derive relationships between concepts of hierarchies [23]. We also intend to explore improvements for the current architecture, e.g., to personalize the DL generator itself as a way of improve efficiency and scalability

References

1.Riecken, D., *Personalized Views of Personalization.* Communications of the ACM, 2000. **43**(4): p. 27-28.
2.Resnick, P. and H. Varian, *Recommender Systems.* Communications of the ACM, 1997. **40**(3): p. 56-58.
3.Fox, E., *et al.*, *Development of a Modern OPAC: From REVTOLC to MARIAN*, in *Proc. 16th Annual Int'l ACM SIGIR Conf. on R&D in Information Retrieval, SIGIR '93*. 1993, ACM Press: Pittsburgh. p. 248-259.
4.Witten, I.H., *et al.*, *Greenstone: A Comprehensive Open-Source Digital Library Software System*, in *Proceedings of the Fifth ACM Conference on Digital Libraries: DL '00, June 2-7, 2000, San Antonio, TX*. 2000, New York. p. 113-121.
5.IBM, *IBM DB2 Digital Library*, 2000, IBM.
6.Ramakrishnan, N., *PIPE: Web Personalization by Partial Evaluation.* IEEE Internet Computing, 2000. **4**(6): p. 21-31.
7.Gonçalves, M.A., *et al.*, *Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries*, 2001, Virginia Tech. Technical report
8.Berry, M.W., S.T. Dumais, and G.W. O'Brien, *Using Linear Algebra for Intelligent Information Retrieval.* SIAM Review, 1995. **37**(4): p. 573-595.
9.Gonçalves, M.A., *Constructing Geographic Digital Libraries using a Hypermedia Framework.* Multimedia Tools and Applications, 1999. **8**(3): p. 341-357.
10.Spiliopoulou, M., *Web Usage Mining for Web Site Evaluation.* Communications of the ACM, 2000. **43**(8): p. 127-134.
11.Fernandéz, M., *et al.*, *Declarative Specification of Web Sites with Strudel.* The VLDB Journal, 2000. **9**(1): p. 38-55.
12.Anupam, V., *et al. Personalizing the Web Using Site Descriptions*, in *DEXA Workshop*. 1999.
13.Abiteboul, S., *et al.*, *Declarative Specification of Electronic Commerce Applications.* IEEE Data Engineering Bulletin, 2000. **23**(1): p. 37-42.
14.Oliveira, J.L.d., M.A. Gonçalves, and C.B. Medeiros, *A Framework for Designing and Implementing the User Interface of a Geographic Digital Library.* International Journal on Digital Libraries, 1999. **2**(3): p. 190-206.
15.Jones, N.D., *An Introduction to Partial Evaluation.* ACM Computing Reviews, 1996. **28**(3): p. 480-503.
16.Gonçalves, M. A., France, R. K., Fox, E. A., *MARIAN: Flexible Interoperability for Federated Digital Libraries*, Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2001), Darmstadt, Germany, September 4-9 2001 (to appear).
17.Gonçalves, M.A., *et al. MARIAN Searching and Querying across Heterogeneous Federated Digital Libraries.* in *Proceedings of the First DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries,* 2000.
18.Nestorov, S., S. Abiteboul, and R. Motwani:, *Inferring Structure in Semistructured Data* Sigmod Record,1997. **26**(4):p. 39-43.
19.Qiu, Y. and H.P. Frei. *Concept Base Query Expansion.* in *Proceedings of the ACM-SIGIR Conference*. 1993.
20.Chakrabarti, S., *et al.*, *Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies.* VLDB Journal, 1998. **7**(1): p. 163-178.
21.Jobson, J.D., *Applied Multivariate Data Analysis.* Vol. II: Categorical and Multivariate Methods. 1992: Springer-Verlag.
22.Binkley, D. and K. Gallagher, *Program Slicing.* Advances in Computers, 1996. **43:** p. 1-50.
23.Sacco, G.M., *Dynamic Taxonomies: A Model for Large Information Bases.* IEEE Transactions on Knowledge and data Engineering, 2000. **12**(3): p. 468-479.