PIMO - a Framework for Representing *Personal Information Models*

Leo Sauermann, Ludger van Elst, Andreas Dengel
Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
{firstname.surname}@dfki.de

Abstract: This paper presents the concept and realization of a *P*ersonal *I*nformation *Mo*del (*PIMO*). A PIMO is used to represent a single users' concepts, such as projects, tasks, contacts, organizations, allowing files, e-mails, and other resources of interest to the user to be categorized. This categorization using multiple criteria was used to integrate information across different applications and file formats. Based on RDF/S, multiple layers were defined: an upper-layer for a minimal set of generic concepts, a mid-layer for refinements, and a user-layer for concepts of the individual user. Our approach was deployed and used in several research projects. The PIMO helps users to categorize resources for Personal Information Management (PIM), it is intended to be the integrative part in personalized systems, such as Social Semantic Desktops.[1]

Key Words: personal information management, ontologies, personalization
Category: H.1.1, H.3.3

1 Motivation

In the EPOS project [6] the use of ontologies was suggested on all levels of the organization, starting with the desktops of individual knowledge workers. Their personal knowledge workspace encompass e-mails, files, contacts, projects and resources from the corporate intranet [2]. Today, products exist to create order on top of these structures, for example mind mapping tools, project management tools, or *Personal Information Management* (PIM) tools. However, the semantics of these structures is typically buried in the individual application.

The core of the EPOS approach is the *Personal Information Model* (*PIMO*). It is a formal representation of the structures and concepts an individual knowledge worker needs, according to her or his personal mental model. It is an application-independent and domain-independent representation. Concepts used to categorise elements in one application will also appear in other applications. Based on studies about file management we know about the importance of these structures for finding and reminding information [3]. The value of the existing

[1] This work was supported by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (Grant 01 IW C01, Project EPOS, and Grant 01 IW F01, Project Mymory) and by the European Union IST fund (Grant FP6-027705, Project NEPOMUK). The authors thank the EPOS project team for input and Laura Zilles for performing the study which led to the DFKI-mid ontology.

structures should be kept and augmented now. We have already developed the model of using multi-perspective classification [7] and how such structures can be used for personalization [13]. There has been research in synchronizing different applications to each other in a n:n approach [4], the PIMO model lowers the costs here because it allows each application to integrate with the PIMO and not to all other applications. For the *Semantic Desktop*, where Semantic Web technologies are already used, the PIMO is a cornerstone for data integration.

2 Definition of a PIMO

Formal structures such as OWL and RDFS ontologies have been used for Personal Information Management before [17], our approach includes a suggestion for a domain model for knowledge work, coined *PIMO-Upper*. Before taking a closer look, we give a definition of terms which are partly based on the definition of TopicMaps [11, 12]. The **Personal Knowledge Workspace** [9] (or "Personal Information Space" [17]) embraces all data "*needed by an individual to perform knowledge work*". It is (1) independent from the way the user accesses the data, (2) independent from the source, format, and author of the data. **Native Resources** are part of the personal knowledge workspace, personal files of the user, e-mails, and other PIM related resources, such as appointments or contacts. In Topic Maps, this is an occurrence that is categorized. **Native Structures** are categorization schemes for Native Resources such as file-system folders, bookmark folders, e-mail folders, tags. In the (common) case that a user operates in a document-centered way his internal representations of these concepts are already largely reflected in content and structuring of his information elements (see, *e. g.*, [7]). There are file folders called "projects", e-mail folders named with costumers' names, product names in file designators, etc. The **Mental Model** is part of the cognitive system of the person. Subjective to the person, the mental model is individual and cannot be externalized thoroughly. The PIMO aims to represent parts of the Mental Model necessary for knowledge work.
Now, a definition for a Personal Information Model can be given.

Definition 1. A PIMO is a Personal Information Model of one person. It is a formal representation of parts of the users Mental Model. Each concept in the Mental Model can be represented using a Thing or a subclass of this class in RDF. Native Resources found in the Personal Knowledge Workspace can be categorized, then they are occurrences of a Thing.

The vision is that a *Personal Information Model* reflects and captures a user's personal knowledge, *e. g.*, about people and their roles, about organizations, processes, things, and so forth, by *providing the vocabulary* (concepts and their relationships) for required expressing it as well as concrete instances. In other words, the domain of a *PIMO* is meant to be "all things and native resources that are in the attention of the user when doing knowledge work". Though "native" information models and structures are widely used, there is still much potential for a more effective and efficient exploitation of the underlying knowledge. We think that, compared to the cognitive representations humans build, there are mainly two shortcomings in native structures:

- *Richness of models*: Current state of cognitive psychology assumes that humans build very rich models, encoding not only detailed factual aspects, but also episodic and situational information. Native structures are mostly taxonomy- or keyword-oriented.
- *Coherence of models*: Though nowadays (business) life is very fragmented humans tend to interpret situations as a coherent whole and have representations of concepts that are comprehensive across contexts. Native structures, on the other hand, often reflect the fragmentation of multiple contexts. They tend to be redundant (i.e., the same concepts at multiple places in multiple native structures). Frequently, inconsistencies are the consequence.

The *PIMO* shall mitigate the shortcomings of native structures by providing a *comprehensive model* on a *sound formal basis*. In the following, we describe how the concept of a*PIMO* has been realized within the EPOS project.

3 Realization of the PIMO

When building concrete *PIMO*s, we now have the problem of two, potentially conflicting demands: On the one hand, we want to give the user the opportunity to span his information space largely in the way *he* wants. The *PIMO* should model *his* mental models. In consequence, we cannot prescribe much of this structure. On the other hand, "empty" systems often suffer from the cold start problem, being not accepted by user when not already equipped with some initial content. Using a multi-layer approach (see also [14]), we try to find a balance through providing the presentational basis as given, which users *can* incorporate or extend:

- PIMO-Basic: defines the basic language constructs. The class pimo-basic:Thing represents a super-class of other classes.
- PIMO-Upper: A domain-independent ontology defining abstract sub-classes of Thing. Such abstract classes are PersonConcept, OrganizationalConcept, LocationConcept, Document, etc.
- PIMO-Mid: More concrete sub-classes of upper-classes. The mid-level ontology serves to integrate various domain ontologies and provides classes for Person, Project, Company, etc.

- Domain ontologies: A set of domain ontologies where each describes a concrete domain of interest of the user. The user's company and its organizational structure may be such a domain, or a shared public ontology. Classes are refinements of PIMO-Mid and PIMO-Upper, allowing an integration of various domain ontologies via the upper layers.
- PIMO-User: the extensions of above models created by an individual for personal use. Classes, properties and things are created by the user.

3.1 The Representational Assumptions: PIMO-Basic

To represent things and their relations, different standards are available. First, RDFS which can represent classes and subclass relationships, properties and subproperty, and resources that are instances of classes and can be described with properties. Second is OWL which integrates reasoning capabilites and description logic. Third, there are standards to describe mind maps, such as the XTM standard. Using rules or description logic is not in the requirements for personal information models, therefore we have selected RDFS as our representation language instead of OWL.

In PIMO-Basic, two key concepts are introduced: the classes *Thing* and *ResourceManifestation*. *Thing* is a superclass of abstract concepts and physical objects, with the aim of representing them on a conceptual level. *ResourceManifestation* is a class to represent the documents in a computer system. The *native structures and resources* can be transformed to RDF as presented in [5],[15]. They are represented using subclasses of *ResourceManifestation*. The separation of Things from ResourceManifestations was missing in OWL and RDFS. Although SKOS models it as separation between concepts and resources, it doesn't reuse RDFS subclass-relations and therefore domain/range restrictions or any typed properties are not usable.

The idea is that a `Thing` can now occur in one or many resources. This is represented by a `occurrence` relation. For example, the city Rome (as a concept) can occur in a website about business in Rome (a document). This relation allows the annotation of documents according to the ontology. Certain occurrences are more tightly bound to a concept, when the topic of the document to describe exactly this concept, we model these as `groundingOccurrence`. A grounding occurrence of the concept of the City of Rome could be the wikipedia page about it. For people, the grounding occurrence of the Person "Paul" could be the address book entry with the contact information about "Paul", for a comany the website of the company. Independent of the application domain, grounding occurrences provide user-readable descriptions of the concept in question, and can be used to automatically map PIMOs of multiple users, when two concepts from different users have the same grounding, chances increase that the concepts are the same. This is comparable to XTM occurrence-references of non-addressable topics.

Besides implicit mapping using occurrences, it is also possible to explicitly map things. For this, the `hasOtherRepresentation` relation is defined. When two things are formalisations of the same concept, they can be mapped using this property. Note that this should not happen inside one user's PIMO but rather when domain ontologies or multiple PIMOs are mapped. Additional to instances, also classes can be mapped. For this, the meta-class `PimoClass` was created as subclasses of RDFS-Class. Using a meta-class allows adding annotations about classes in a clean way. Mapping classes is realized with the `hasOtherConceptualization` relation.

3.2 DFKI-KM-Mid: Acquisition of an Exemplary PIMO Mid-Level

The upper level of a *PIMO* just makes a few, basic ontological statements about things which exist on a Semantic Desktop, *i. e.*, things which are essential in a knowledge worker's mental model: *Information elements*, *people-*, *organization-* and *process-related things*, but of course also basic ontological categories like *space* and *time* concepts well-known (and imported) from other typical upper-level ontologies. Obviously, the commitment in this statement is very fundamental for the concept of a Semantic Desktop, but also very abstract. In order to avoid a *cold start problem*[2] with PIMO-based applications, we pre-modeled a PIMO-Mid-Level as a refinement of the upper level which serves two purposes: Firstly, the concepts of the mid level serve as *anchor points* for a user's personal incremental extensions of his PIMO. For example, having already a couple of *project types* as examples in his PIMO (instead of just having projects as abstract organizational concepts) makes it probably much easier for him to classify already existing projects or to model new project types. Moreover, offering a common mid level layer to a *group of people* can also be seen as a *seed for a shared conzeptualization* between these people, facilitating information exchange on the basis of these shared parts of their PIMOs. So, conceptually, the *scope* of a PIMO mid-level is a *group of user's* who potentially share many concepts on their Semantic Desktop (*e. g.*, people in the same department), while the control with respect to extensions or modifications is intended to be at the individual user.

In our prototype, we modeled an exemplary PIMO mid-level using the following methodology, consisting of the three phases *seeding*, *reality match*, and *evolution*: In the seeding phase, a couple of exemplary *native structures* (file and email folders) of members of DFKI's Knowledge Management Department were manually analyzed and so laid the basis for an *initial DFKI-KM-Mid model*.

[2] The problem of cold starts is very well known in knowledge-based systems: In the beginning a system, like a shell, just has little of no information and therefore seems not to be useful to a new user. Consequently, he is not motivated to invest in using and feeding the system with new information which would be a prerequisite to be *more* useful.

DFKI-KM-mid mainly consisted of concepts without deeper modeling, like attached slots etc. In the second phase, this initial model was checked by a detailed survey. 23 members of the department were interviewed whether the initial model fit their individual native structures, which concepts were missing in the model or not occurring in their native structures The results from the reality match were used for evolving and extending the DFKI-KM-Mid model. Further extensions have been made by a more detailed modeling of slots and by the integration of third-party ontologies like FOAF and specially tailored domain ontologies like the "*Organizational Repository*", formalizing the employees and projects of the DFKI KM lab.

Here the idea is that when bringing the *PIMO* idea into a specific environment the mid level should be re-modeled in a similar way as described above. [16] shows an example for that in a concrete business scenario.

4 Applied use of the PIMO

Using above prerequisites, the *Personal Information Model* of a user can now be created by assembling the different parts. We will use the example user *Paul* and **Paul's PIMO**[3]. The following steps are necessary: Firstly, PIMO-Basic, PIMO-Upper, PIMO-Mid are imported unchanged Then, one or more domain ontologies are imported, *e. g.*, the "Organizational Repository" of a company. The personal mental model of the user is represented in the user's own domain ontology, called PIMO-User. The user works within his own namespace, abbreviated using `paul:`. The first element is the user himself, `paul:Paul`. He is represented as instance of the class `pimo:Person` and annotated as owner of his PIMO, which is represented as `paul:PaulsPim`. The user can refine existing classes by creating subclasses and instances. Finally, the native resources on the desktop of the user (files, e-mails, address-book, etc) are converted to data vocabularies using adapters. They are matched to the personal mental model and to domain ontologies.

Hence, the *Personal Information Model* (PIMO) of a user can be defined as the sum of imported upper and mid-level ontologies, domain ontologies, one personal mental model of the user (PIMO-User), and the native resources found in heterogenous data sources.

As an example for a project managed by Paul, we assume he is planning to open a branch office of his company in Rome, Italy. This project is represented as `paul:BranchOfficeRome`, an instance of class `pimo:Project`. To express that the co-worker Tim is part of the project, `paul:Tim` was created and related to the project via the `pimo:hasPart` relation. Tim has a grounding occurrence in the address book of Paul, the address book entry is a resource manifestation.

[3] The models can be retrieved from: http://ontologies.opendfki.de/repos/ontologies/pim/pauls-pimo.pprj

The example goes on to create a custom class (`paul:BusinessPlan`) and custom properties (`paul:manager`).

The PIMO was used as a basis for the EPOS project, and the possibilities to expand it and customize it to certain scenarios is decribed in [13]. It is also the basis for data representation in the *Gnowsis* project, which was decribed in [14].

Norberto Fernandez created an approach to populate a PIMO while the user is doing search tasks. The user interface of his SQAPS search engine automatically creates PIMO concepts in the background, annotating them with Wikipedia pages [8] as grounding resources.

5 Related Work

A similar approach was used by Huiyong Xiao and Isabel F. Cruz in their paper on "A Multi-Ontology Approach for Personal Information Management", where they differentiate between *Application Layer, Domain Layer and Resource Layer*. Alexakos et al. described "A Multilayer Ontology Scheme for Integrated Searching in Distributed Hypermedia" in [1]. There, the layers consist of an *upper search ontology layer, domain description ontologies layer, and a semantic metadata layer*.

PIMO is different from XML Topic Maps (XTM) as it allows to use inference and RDFS definitions, also enabling an efficient way to store the data in RDF databases (whereas XTM is based on XML). The main difference to RDF is that Topic Maps Associations are by definition n-ary relations, whereas in RDF the relations are typically binary. In RDF, a similar approach as to XTM is the SKOS vocabulary [10]. It represents all Things using the class *Concept*, this blocks reusing inference and typed properties of concepts (like the "first name" property of a person cannot be modelled in SKOS).

6 Summary and Outlook

In this paper we presented the *Personal Information Model* — PIMO. It is a framework of multiple ontologies to represent concepts and documents that are in the attention of the user when doing knowledge work. Basic concepts such as time, place, people, organizations, and tasks are pre-modelled in a mid-level ontology that can be extended by the user at will, to express their mental model. Items can be assigned to multiple concepts, extending the limitations of current hierarchical file system. The PIMO was used in the EPOS, Gnowsis, and SQAPS research projects, *e. g.*, for personalization [13] and semantic retrieval services.

Future challenges are in refining the upper and mid-level models, based on experiences gained through evaluations within the NEPOMUK project. There, the PIMO will be used as a means for file and e-mail annotation in various software applications, for example in the Linux KDE desktop. More research needs to be directed towards automatically creating PIMO structures based on analysing native resources and structures.

References

1. C. Alexakos, B. Vassiliadis, K. Votis, and S. Likothanassis. A multilayer ontology scheme for integrated searching in distributed hypermedia. In S. Sirmakessis, editor, *Adaptive and Personalized Semantic Web*, number 14 in Studies in Computational Intelligence. Springer, 2006.
2. J.-T. Bähr, L. van Elst, A. Lauer, H. Maus, L. Sauermann, and S. Schwarz. EPOS – Guiding Example. internal report, DFKI, 2004.
3. D. Barreau and B. A. Nardi. Finding and reminding: File organization from the desktop. 1995.
4. R. Boardman. *Improving Tool Support for Personal InformationManagement*. PhD thesis, Department of Electrical and Electronic Engineering Imperial College London University of London, July 13 2004.
5. A. S. C. Bizer. D2rq-treating non-rdf databases as virtual rdf graphs. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*, 2004.
6. A. Dengel, A. Abecker, J.-T. Bähr, A. Bernardi, P. Dannenmann, L. Elst, S. Klink, H. Maus, S. Schwarz, and M. Sintek. EPOS – Evolving Personal to Organizational Knowledge Spaces, 2002.
7. A. R. Dengel. Six thousand words about multi-perspective personal document management. In *Proc. EDM IEEE Workshop*. IEEE, Oct 2006.
8. N. Fernandez-Garcia, L. Sauermann, L. Sanchez, and A. Bernardi. Pimo population and semantic annotation for the gnowsis semantic desktop. In *Proceedings of the Semantic Desktop and Social Semantic Collaboration Workshop at the ISWC*, volume 202 of *CEUR-WS*, 2006.
9. H. Holz, H. Maus, A. Bernardi, and O. Rostanin. From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management. volume 0, pages 101–127, 2005.
10. A. Miles (Ed.). Simple knowledge organisation system (skos). Technical report, Feb 2004.
11. S. Pepper and G. Moore (Eds.). Xml topic maps (xtm) 1.0. Specification, TopicMaps.Org, 2001.
12. H. Rath. The topic maps handbook detailed description of the standard and practical guidelines for using it in knowledge management. empolis white paper, empolis GmbH, 2003.
13. L. Sauermann, A. Dengel, L. van Elst, A. Lauer, H. Maus, and S. Schwarz. Personalization in the epos project. In *Proceedings of the Semantic Web Personalization Workshop at the ESWC 2006 Conference*, pages 42 – 52, 2006.
14. L. Sauermann, G. A. Grimnes, M. Kiesel, C. Fluit, H. Maus, D. Heim, D. Nadeem, B. Horak, and A. Dengel. Semantic desktop 2.0: The gnowsis experience. In *Proc. of the ISWC Conference*, pages 887–900, Nov 2006.
15. L. Sauermann and S. Schwarz. Gnowsis adapter framework: Treating structured data sources as virtual rdf graphs. In *Proceedings of the ISWC 2005*, 2005.
16. M. Siebert, P. Smits, L. Sauermann, and A. Dengel. Increasing search quality with the semantic desktop in proposal development. In *Proceedings of the Practical Aspects of Knowledge Management PAKM conference*, volume 4333/2006 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin / Heidelberg, 2006.
17. H. Xiao and I. F. Cruz. A multi-ontology approach for personal information management. In S. Decker, J. Park, D. Quan, and L. Sauermann, editors, *Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6*, volume 175, November 2005.