# Personalization in the EPOS project

Leo Sauermann, Andreas Dengel, Ludger van Elst, Andreas Lauer, Heiko Maus, Sven Schwarz

Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
Kaiserslautern, Germany
{firstname.surname}@dfki.de

**Abstract.** In this work we present the results of the EPOS project with regard to the needs of personalization in the Semantic Web. Focus of this work is the subjective view of an individual person, expressed in a *Personal Information Model (PIMO)*. It is matched both with personal resources (files, e-mails, and websites) of the user and organizational knowledge (ontologies). A user observation component gathers actions of the user to calculate the current context with regards to current goals and matching elements in the user's PIMO. Combined, the representation of the user's stored information and the current context provide a thorough representation of the user. Desktop applications can use this representation to provide personalized services. Three special purpose applications were implemented: a search engine, a context-sensitive assistant, and a tool for filing new information. An evaluation of this approach showed that it increases productivity and indeed reflects the subjective view of users. Also, the approach satisfies most of the requirements of an Adaptive Educational Hypermedia System *AEHS*. Parts of this work are published as open source projects. .[1]

## 1 Introduction

As Baldoni et al. described in [1]: personalization in the web requires semantic data markup. The same can be said for the desktop computers: we miss the markup.

A prerequisite for personalization is a representation information about the person itself and about the current work context of the person. On desktop computers, we miss the semantic markup of resources like files, e-mails, or browsed websites. Additionally, we miss the representation of the user's current work context, what resources is the user accessing at the moment and what are the goals of the user. On the Semantic Desktop as described in [13, 5] these possibilities exist. The EPOS project is such a Semantic Desktop and addresses these two problems, providing the following solutions:

[1] This work was supported by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (Grant 01 IW C01, Project EPOS: Evolving Personal to Organizational Memories).

– A representation of the user's personal information items, including e-mails, files, and other data sources using RDF
– A representation of the user's mental model in a formal representation, using several layered ontologies.
– A desktop service to capture the current actions of the user, representing the actions using RDF and then calculating the current context of the user.

In this paper we will give an introduction of the EPOS scenario and describe the solutions relevant for personalization. Note that we reference elements from the OWL and RDF/S specifications without introducing them [3, 9].

### 1.1 The scenario of EPOS

EPOS aims at supporting knowledge workers doing knowledge-intensive tasks. As an example, a consultant within a German IT company is investigated. The knowledge worker handles information from various information sources, local data sources like files stored on his desktop computer or contacts in his local address-book, organizational data from shared file repositories or ontologies and web resources. While the organization asks for universally applicable and standardized persistent structures, processes, and work organizations to achieve and maintain universally accessible information archives, the individual knowledge worker requests individualized structures and flexibility in processes and work organization in order to reach optimal support for the individual activities.

A problem of conventional file-systems (for example on the windows operating system) is that they do not allow the user to place the same file into two folders, if the folders represent topics and the files represent articles, then the user cannot express the fact that an article covers two or more topics using these structures. A solution to this problem can be reached by using a multi-criterial classification approach (placing the same file in multiple classes), which is possible using graph models like RDF. In EPOS, we allow the user to classify a file stored on the local hard disk by connecting it to one or more entities in the users's Personal Information Model (PIMO).

## 2 The Personal Information Model (PIMO)

The *Personal Information Model* as created in EPOS was driven by four requirements:

– Sound formal basis: The PIMO must support various knowledge services, among them logics-based services (e.g., ontology-based information retrieval). Therefore, the PIMO must employ an expressive representation language and has to wipe out the contradictions and redundancies of the native structures.
– Bridge between individual and organizational Knowledge Management: The PIMO has to incorporate global ontologies, but also has to reflect the changes and updates of native structures. The PIMO itself should be a source of input for OM-wide ontologies.

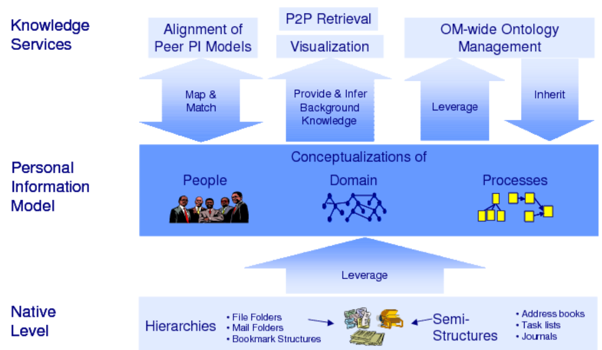


**Fig. 1.** The Personal Information Model as *semantic middleware* between native structures and knowledge services.

– Maintenance: Adequate means have to be provided that assist the user with stepwise formalization of native structures and inspection of the PIMO.

Figure 1 shows how such a Personal Information Model is embedded in the EPOS information landscape and what basic functionality has to be provided to link the PIMO to the native structures as well as to the envisioned knowledge services. The central part of the PIMO are formal representations of mental models, namely people, concepts, processes, etc. Native resources (files, e-mails, web documents) are expressed using a separate ontology and according services to transform the native data into this ontology. The services for peer-to-peer ontology alignment and organizational-memory (OM) wide ontology management are described in [19].

The PIMO ontology framework created in the EPOS project consists of six components, see figure 2. The first half of these components represent mental models on a conceptual level using formalized domain ontologies. Itself it consists of three layers: upper-level, mid-level and domain ontologies. Native resources (files, e-mails, web-pages) are represented using native data vocabularies. For example, the mental model of the person "Heiko Maus" is formally represented in many native resources, the address book entry of Heiko Maus is a native representation of the mental concept of the living person, which is expressed in a domain ontology or personal model.

### 2.1 Native structures and data

The personal workspace with its *native structures* like file- and mail-folder hierarchies reflects the worker's personal view of his or her information space. The underlying conceptualizations are therefore a valuable aid not only to guide the worker's information management tasks like storage and retrieval, but also to the internalization and, ultimately, utilization of new information. Furthermore, due

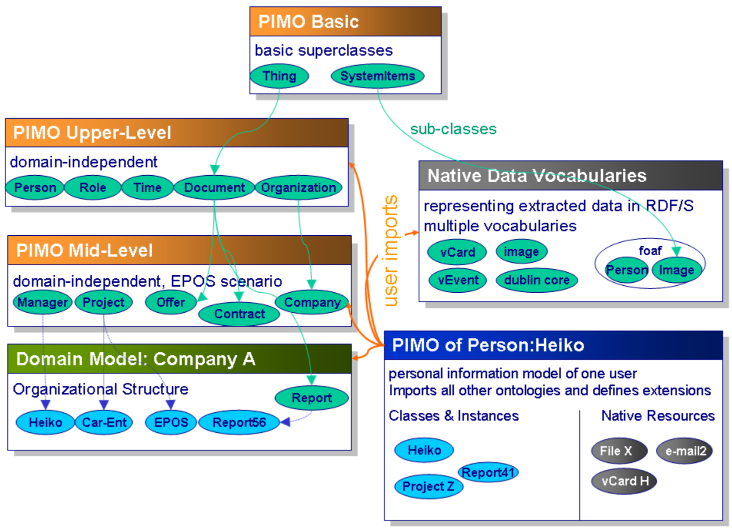


**Fig. 2.** PIMO ontology components

to their continuous development by the knowledge workers, the *personal* structures provide an excellent input for the acquisition of *organizational* knowledge. However, today's native structures also have some serious drawbacks. They are often built ad hoc, which means they only reflect a snapshot of the worker's view. They lack formal semantics. Therefore, they are hard to exploit by automatic information services.

A framework was created to transform several native structures to the RDF format, this was implemented as part of the gnowsis semantic desktop[12]. By the end of 2005, we have implemented adapters for Microsoft Outlook (e-mails, appointments, address book), IMAP e-mail servers, the Thunderbird address book, several file formats and other applications. External contributors have commited adapers to the Flickr.com photo website (by Anja Jetzsch and Florian Mittag) or relational databases (by Richard Cyganiak). Our approach to extracting the data from native applications using *virtual rdf graphs* was published in[14]. To improve the performance of this approach and to open our results to a wider community, the code for data extraction was moved to the Aperture project on sourceforge [2]. In EPOS, we focused on these data sources of native structures:

- several file formats (bibtex, Microsoft Office, PDF, etc.)
- Microsoft Outlook items (calendar, address book, e-mails, todo list)
- IMAP e-mails
- web-sites browsed in the Firefox browser

[2] (http:
aperture.sourceforge.net)

– weblog entries in the Wordpress blogging system

The data extracted from these native sources is then described using a set of data-oriented RDF/S vocabularies, represented in the layer of *Native Data Vocabularies*. These were selected during creation of the adapter software by the adapter developers. For example, to represent address book items from Microsoft Outlook a vCard[3] RDF/S representation was taken. Note that these vocabularies were matched to the other PIMO ontologies using subclass relations. The relations between native resources and other concepts in the ontology layers can be compared to occurrence-relations in Topic Maps[18].

### 2.2 PIMO-Basic, PIMO-Upper, PIMO-Mid and Domain Ontologies

Apart from the native structures, the mental models are represented in EPOS using a multi-layer approach. 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 [4]. There, the layers consist of an *upper search ontology layer, domain description ontologies layer, and a semantic metadata layer.*

In EPOS, the ontology layers consist of

– 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 EPOS 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. Domain ontologies should sub-class from PIMO-Mid and PIMO-Upper to allow integration.
– PIMO-User: the extensions of above models created by an individual for personal use.

The first three layers were created once by members of the EPOS team and are well suited and quite fix for such knowledge work scenarios, the domain ontologies are created for real domains and change frequently. For example, a domain ontology was created to represent the organizational structures at the DFKI KM lab, named "*Organizational Repository*".

[3] http://www.w3.org/TR/vcard-rdf

### 2.3 The PIMO of an individual user

Using above prerequisites, the *Personal Information Model* of a user can now be created by assembling the different parts for a single user.

- PIMO-Basic, PIMO-Upper, PIMO-Mid are imported unchanged
- One or more domain ontologies are imported. The "Organizational Repository" of a company can be represented as domain ontology.
- The personal mental model of the user is represented in the user's own domain ontology, called PIMO-User. There personal concepts, ideas, projects, contacts etc. are represented and matched to domain ontologies. The user can create his own classes and instances here.
- 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. It fulfills the requirements stated above: formal basis, bridge between personal and organizational knowledge, maintenance.

Mappings between ontologies were either realized by using subclass/subproperty relations to map classes or by using the custom property pimo:hasOtherRepresentation to express the fact that one instance A1 of ontology O1 is represented in another ontology O2 in the instance A2. A1 pimo:hasOtherRepresentation A2 would be the according triple. These mappings were primarily used to match instances created by the individual user in his individual model to instances in domain ontologies. For example, the user creates an instance for the project "Car-Entertainment" and later connects it to the instance in the organizational repository representing the same project. As mentioned above, automatic algorithms to do this were evaluated in [19].

Based on the PIMO of a user, we can now look at the activities of the user, from which we can derive a model of the user's context.

### 2.4 User activity and user modeling

In [16] we presented our approach to capture user activity and represent the user's context and goals.

We identified four different levels of abstraction of user activity: The first level, called *Workspace Level*, represents the operating system and the applications that provide access to files, objects and information structures. Observation at this level results in workspace events such as various mouse clicks, entering of some text, or starting and handling applications. Data handled in this interaction is described in the *Native Data Vocabularies*. The *User Action Level* contains such user actions as "create new text document", rather than atomic mouse-clicks or actions like start text editor or activate File-new menu. Those user actions will be inferred from a series of workspace events described before.

While he interacts with the computer, he always has some higher medium term goal in mind such as compiling an offer to a customer or writing a project proposal. The documents handled can be represented in the PIMO ontologies. Those user goals are captured in the *Task Concept Level* and are represented by task concepts which are concepts in an ontology about such user goals. EPOS will elicit the users goal(s) from a sequence of the user actions needed to be carried out to achieve this goal. And, last but not least, the *Process Level* connects to the organizational structures processes which are explicitly modelled in the domain ontology representing the company. If there is a Workflow-Management System (WfMS) available that also interprets the company ontology, we can connect / assign the user to running workflows. Workflows can be semantically described using the same set of task concepts as were eliciting from the users behavior [15]. In EPOS, the integration with the WfMS was realized by a gnowsis adapter [17].

### 2.5 Relation to existing approaches

The EPOS system as whole is an implementation of a Semantic Desktop, as defined in [13]. Competing semantic desktop implementations, like Haystack, are also discussed there. It is data and application-centric but misses the requirements of personalization, which we find in [1]. In this work titled "*Personalization for the Semantic Web*" we find a formalism for an *Adaptive Educational Hypermedia System* (AEHS). An AEHS is a Quadruple of Document Space *DOCS*, User Model *UM*, Observations *OBS*, Adaptation Component *AC*.

Via the EPOS context elicitation system, the requirements of UM (User Model) and OBS (Observations) are satisfied. The UM can be either seen as the sum of all data represented in one user's PIMO or only by taking the elicitated context. The OBS requirement is satisfied by the user actions that we described in short above and in detail in [16, 15]. Rules were used in all stages of the architecture. A good example for the usage of semantic web rules for personalization is the gnowsis desktop search, which also fulfills the AC requirement.

## 3 Personalized special purpose applications

Given the formal representation of the user model in the PIMO framework different use cases were identified in the EPOS project. Three such use cases and their respective implementation will be described in this paper, in all three cases, the PIMO framework supplies the ontology data.

### 3.1 A drop-box for filing

As identified by Indratmo and Cruz[7] and earlier Barreau and Nardi[2], filing information is a crucial task in personal information management. The EPOS solution is to provide a special folder for the user, called *Drop-Box*, where the user can file information. The name is derived from the Mac OS drop-box. When the knowledge worker downloads a new file from the internet, it is stored into the special drop-box folder. EPOS analyzes the content of the file and searches for documents that are similar to the file (using the *Brainfiler* text classification system by Brainbot AG). The documents again match to concepts of the user which again are mapped to the domain ontologies. Note that the training of documents to concepts has to be done beforehand. The result of this matching is that the system can suggest which elements of the PIMO of the user match against the dropped file. The user selects which concepts are really describing the file and the file is moved to a folder that is related to one of the concepts. This approach allows the user to use multiple classifications, breaking with hierarchical file structures. By using the drop-box frequently, the categories are further trained.

### 3.2 Gnowsis desktop search

For information retrieval, a desktop search tool was created, called *gnowsis desktop search*. Its main input is a fulltext-search field as known from common desktop search tools (google desktop). The search reaches across all parts of the PIMO framework, domain ontologies, native data sources and the PIMO-User. Internally, the search engine represents the search results in a RDF graph model, inspired by the Roodolf RDFS model for the Google api[4]. Our vocabulary contains classes to represent the search request, each returned hit, and more, see here[5]. Representing the search results as RDF before rendering them as HTML output allows EPOS to run personalization rules before rendering the results. These rules are expressed using the Jena[8] Rule syntax as described in the Jena documentation. We also embedded the possibility to call additional SPARQL[10] queries from within the rule engine, hence a rule can decide to expand the search by invoking another search to the ontologies. Personalized sets of these rules can be used to expand the search results (increasing recall values) or to filter out unwanted results (increasing precision). In the EPOS scenario, rules were used to include defined ontology mappings (hasOtherRepresentation links, see above). This is a shortened version of the rule set used in the evaluation of the system:

```
# found something? -> infer other representations via SPARQL\\
(?hit retrieve:item ?x) -> \\
querySparql('CONSTRUCT { ?x pimbasic:hasOtherRepresentation ?y } ')

# found a project? -> also show members \\
(?hit retrieve:item ?project), (?project rdf:type org:Project) ->\\
querySparql('CONSTRUCT   {
  ?project org:containsMember ?m.
}).
```

[4] http://nutria.cs.tu-berlin.de/roodolf/rdfs
[5] http://www.gnowsis.org/ont/gnoretrieve

### 3.3 Context-sensitive assistance system

This assistant system should react to the current work context of the user (topic, content, project, etc.) , providing help that matches the current goal of the user. It should provide help to the user without explicit invocation of it by the user (*pro-active*). Also, the user is not always aware that assistance is available and therefore the assistant has to work in a fashion that is not disturbing the user, disrupting the user or distracting the user (*non-intrusive*). The approach is similar to the Lumière project[6] by Microsoft Research, which in part was implemented in the Microsoft Office assistant. Also Rhodes' "Margin Notes" project[11] aims at a similar goal: while the user is web-browsing, margin notes loads relevant information from the local files of the user and shows them in a side-bar in the browser.

The EPOS assistant is implemented in a sidebar component. The user interface of the EPOS assistant can be shown or hidden, as the user wishes. The sidebar contains also other components like a desktop search field and an ontology overview, so the assistant is not taking much screen space. It updates itself when the user context model changes based on the user actions (as described above). When the user accesses a resource (for example a web-resource in the browser), related concepts are identified (using text analysis of the resource) and loaded. The assistant then shows only the relevant documents, people, projects and tasks that are related to the current work context. Again, these concepts are taken from the PIMO-Mid ontology level. The assistant therefore benefits from the integration realized in the PIMO framework. It is *pro-active* because it reacts to context changes detected automatically by the context observation service and it is *non-intrusive* because the shown information relates to the current work context and the decent screen space that is occupied.

## 4 Evaluation of the approach

The EPOS project was evaluated using different methods. First Mark Siebert and Pierre Smits of Siemens Business Services (SBS) adapted the gnowsis desktop search system and compared it to other systems. The rule system provided by the gnowsis desktop search was adapted by them to a scenario at SBS, changing the rules so that they return documents based on a search situation the user is currently in. The results of this evaluation will be published 2006. Then a case study was done where eight knowledge workers at DFKI were observed and questioned over a longer period. Their feedback was collected via two ways: First, a daily interview (questionnaire) provided subjective, qualitative statements, and, second, measuring click counts and explicit user feedback during a one week period delivered qualitative measures. Results of the case-study were:

- The *drop-box* component increased productivity as it allows to file items faster than without the assistance.
- The possibility to add multiple categories to a document was used, in the mean 2.5 categories were attached to a file, which is significantly more than the single category a hierarchical file system provides.

– The *gnowsis desktop search* was used very frequently. Users answered in the questionnaire that they found unexpected information and that the categories provided by the PIMO helped during retrieval.
– The subjects stated, that the context-sensitive assistance came up with unexpected, surprising information items (e.g., documents) revealing new, useful, cross-references.
– The participants agreed that the PIMO reflects their personal mental models.

## 5 Conclusions and Outlook

The EPOS project addressed the problem of knowledge management in heterogenous environments. Personalization services for desktop applications need semantic markup of the resources found on the desktop pc and on the web. For the scenario of knowledge work in the consulting business, EPOS and the gnowsis framework provide a layered ontology solution, the PIMO framework. It represents information with focus on the subjective view of the user. The activity of the user is observed and captured in a detailed user context model[15]. This model can be retrieved and used for assistance, as shown in the pro-active context-sensitive assistant system. Two other special purpose applications were presented, a drop-box for filing information, and a desktop search system that allows a rule-based expansion of search results. An evaluation of these three components showed that they can improve the productivity of knowledge workers. Parts of EPOS match to components needed for an AEHS and we suggest that our approach can assist Semantic Web personalization as such. Implementations of our work are published as open source projects at www.opendfki.de and will be continued in follow up projects.

The architecture of EPOS and the PIMO framework provide building blocks that enable personalization on the Semantic Desktop. The EU project NEPOMUK aims at standardizing the Semantic Desktop platform and the results of EPOS contribute to this effort. The PIMO ontology as it was developed for EPOS is improved and extended on a regular basis and can be downloaded[6]. Currently, parts of the system are re-implemented under the codename "gnowsis beta", a pre-release was already published in March 2006 and a more stable release can be presented at the ESWC2006 workshop.

**Acknowledgements** We want to thank Mark Siebert and Pierre Smits who conducted the evaluation of the gnowsis desktop search at Siemens and have helped to keep the system installable and simple. Richard Cyganiak and Anja Jentzsch have accompanies the gnowsis work ever since. Special thanks are directed to our student contributors (in random order): Benjamin Horak, Frank Osterfeld, Dominik Heim, Daniel Bahls, Daniel Burkhart, Günther Noak, Ralf Biedert, Moritz Plößl, Florian Mittag, Emmanuel Gasne, and many more.

[6] http://ontologies.opendfki.de/repos/ontologies/pim/

## References


1. M. Baldoni, C. Baroglio, and N. E. Henze. *Personalization for the Semantic Web*, volume 3564, chapter p. 173, pages 173–212. Springer, Aug 2005.
2. D. Barreau and B. A. Nardi. Finding and reminding: File organization from the desktop. 1995.
3. D. Brickley and R. Guha. Rdf vocabulary description language 1.0: Rdf schema. w3c recommendation 10 february 2004. http://www.w3.org/TR/rdf-schema/.
4. K. V. C. Alexakos, B. Vassiliadis and S. Likothanassis. *A Multilayer Ontology Scheme for Integrated Searching in Distributed Hypermedia*. Springer, 2006.
5. S. Decker, J. Park, D. Quan, and L. Sauermann, editors. *Proc. of Semantic Desktop Workshop at ISWC*, volume 175 of *CEUR Workshop Proceedings*, 2005.
6. E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommeltse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. Microsoft Research.
7. Indratmo and J. Vassileva. Human and social aspects of decentralized knowledge communities. In *Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6*, 2005.
8. B. McBride. Jena: Implementing the rdf model and syntax specification. In *Proc. of the Semantic Web Workshop WWW*, 2001.
9. D. L. McGuinness and F. Harmelen. Owl web ontology language overview w3c recommendation 10 february 2004. http://www.w3.org/TR/owl-features/.
10. E. Prud'hommeaux and A. S. (edts). SPARQL query language for RDF. W3c working draft, W3C, 2005. http://www.w3.org/TR/rdf-sparql-query/.
11. B. J. Rhodes. Margin notes: Building a contextually aware associative memory. In *Proceedings of the International Conference on Intelligent User Interfaces(IUI2000)*, January 2000. Software Agents Group, MIT Media Lab.
12. L. Sauermann. The gnowsis-using semantic web technologies to build a semantic desktop. Diploma thesis, Technical University of Vienna, 2003.
13. L. Sauermann, A. Bernardi, and A. Dengel. Overview and outlook on the semantic desktop. In *Proc. of the 1st Workshop on The Semantic Desktop at ISWC*, 2005.
14. L. Sauermann and S. Schwarz. Gnowsis adapter framework: Treating structured data sources as virtual rdf graphs. In *Proceedings of the ISWC 2005*, 2005.
15. S. Schwarz. A context model for personal knowledge management. In *Proceedings of the IJCAII WS. on Modeling and Retrieval of Context*, Edinburgh, 2005.
16. S. Schwarz and T. Roth-Berghofer. Towards goal elicitation by user observation. In *Proceedings of the FGWM WS. on Knowledge and Experience Management*, Karlsruhe, 2003.
17. R. Shkundina and S. Schwarz. A similarity measure for task contexts. In *Proc. of the WS Similarities - Processes - Workflows, ICCBR 2005*, Chicago, 2005.
18. G. M. e. Steve Pepper. Xml topic maps (xtm) 1.0. Specification, TopicMaps.Org, 2001.
19. L. van Elst and M. Kiesel. Generating and integrating evidence for ontology mappings. In *Proc. of EKAW 2004*, 2004.