

# Integration and Semantic Enrichment of Explicit Knowledge through a Multimedia, Multi-source, Metadata-based Knowledge Artefact Repository

**Felix Mödritscher**

(Institute for Information Systems and Computer Media,  
Graz University of Technology, Austria, fmoedrit@iicm.edu)

**Robert Hoffmann**

(Institute for Information Systems and Computer Media,  
Graz University of Technology, Austria, rhoff@iicm.edu)

**Werner Klieber**

(Know-Center Graz, Austria,  
wklieber@know-center.at)

**Abstract:** Explicit knowledge is often stored in various repositories within a company. As APOSDLE aims at considering the whole intellectual capital of companies, we are coping with the task to integrate the content of several and different repositories within the APOSDLE platform. Further, other modules in the APOSDLE system also require functionality to semantically enrich knowledge artefacts like documents or multimedia objects. In this paper, we present our approach to these challenges, namely the “Multimedia, Multi-source, Metadata-based Knowledge Artefact Repository”. After presenting the basic idea and critical issues of this solution, we point out related technological approaches as well as their limitations. Finally, we report about a first prototype realising this idea and our experiences gained so far.

**Keywords:** Digital Object Repository, Data Layer, Knowledge Artefact, Metadata Standards

**Categories:** H.2.4, H.2.5, H.2.8, H.3.2, H.3.7, H.5.1

## 1 Introduction

Companies face the problem of continuous growth and increasing amounts of knowledge relevant for their core businesses [Borghoff and Pareschi, 97]. Particularly, explicit knowledge, by means of reports, articles, manuals, patents, images, video, software, and the forth, underlies a high degree of fluctuation. As key resources are often hidden in different heterogeneous knowledge repositories [Dzbor et al., 00], many companies already focus on managing and integrating these repositories into a knowledge management system in order to provide access their contents [Kühn and Abecker, 97].

One of the key issues in the APOSDLE project [APOSDLE, 07] comprises the idea to consider all parts of a company’s intellectual capital and make it available to the knowledge workers. On the one hand, implicit knowledge should be accessible via tools recommending experts within the working context and supplying collaborative features to support knowledge exchange. On the other hand, the documents relevant for a certain task should be captured by the APOSDLE platform and utilised by means of different application scenarios, like providing the knowledge workers with these resources, embedding them into learning events or making discussions possible.

Against this background, this paper introduces our idea of a “Multimedia, Multi-source, Metadata-based Knowledge Artefact Repository” (M3KAR), which ought to be part of APOSDLE’s underlying knowledge infrastructure and deals with managing so-called Knowledge Artefacts. In context of the APOSDLE system, M3KAR aims at providing the functions which allow other modules to semantically enrich digital resources and exploit this semantics. In the following the basic concept of our solution approach is described and aspects relevant for practice are outlined. Thereafter, an overview of related systemic types is given, and differences to M3KAR are highlighted. Finally, the implementation of our idea within the first APOSDLE prototype is described, and experiences are summarised.

## 2 The Basic Concept of M3KAR

As mentioned before, our task in the APOSDLE project is to guarantee a homogeneous access to all documents stored in different repositories within a company. Further, different modules and tools within the APOSDLE platform should also have the possibility to retrieve these documents on basis of a unique identifier, add or modify semantics to the digital objects and utilise this meta-information or the content for their purposes. Therefore, two important issues have to be considered: (1) the Knowledge Artefacts themselves and (2) the systemic architecture of M3KAR.

### 2.1 Knowledge Artefacts and Their Lifecycle

Our first definition involves the so-called Knowledge Artefact (KA) which can be understood as a piece of (digital) information relevant for a certain working context and enriched with semantic information in terms of metadata (see [Ley, 06]). Regarding related fields like standardisation of digital objects (e.g. with the Dublin Core Metadata Element Set) or content management, Knowledge Artefacts underlie a lifecycle within the APOSDLE platform, as shown in Figure 1.

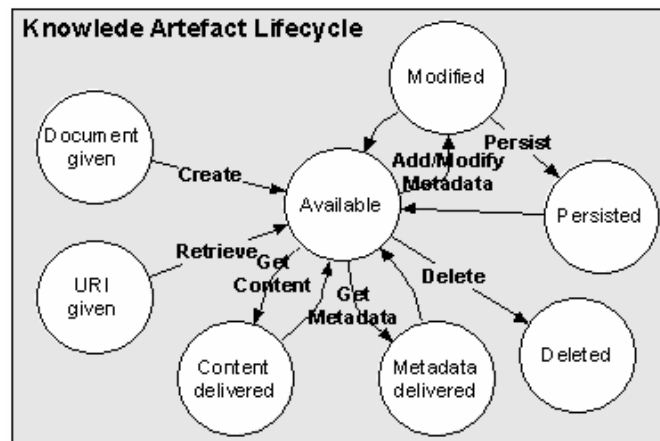


Figure 1: The Knowledge Artefact Lifecycle

Concerning a company's intellectual capital, the following scenarios are of relevance for M3KAR. On the one hand, Knowledge Artefacts might be available in some repository and, therefore, might be retrieved by unique identifiers. On the other hand, KAs can be also created within certain APOSDLE scenarios, e.g. during learning or collaboration. After retrieving or creating a KA, it is made available for other systemic components. The tools and modules can now access the content and metadata of a KA, add or modify attributes or even delete it. As a consequence of this lifecycle, the following six issues are of importance for realising a Knowledge Artefact Repository:

1. Due to the distributed architecture of APOSDLE, a Knowledge Artefact can be considered as unique within the whole platform. Addressing design patterns, [Pant and Ondo, 02] outline the special challenges of the Singleton pattern in distributed environments.
2. The main application scenario for M3KAR comprises the retrieval of Knowledge Artefacts, whereby high-level retrieval methods are provided by another module. Thus retrieval is restricted to the principles of data retrieval, e.g. by retrieving KAs according to their identifier or supplying fuzzy queries on certain attributes to locate the relevant document within the integrated repositories. Therefore, KAs must provide a set of system-managed and read-only attributes, such as a unique URI (Uniform Resource Identifier), in order to be accessible. Additionally, certain metadata fields have to be searchable.
3. Furthermore, other components should be able to add or modify metadata attributes of Knowledge Artefacts. Besides, it is intended that some modules might even create whole KAs within the platform. Thus, changes of the KA need automatically to be persisted. Moreover, also the modifications of the original content within the repositories have to be detected and regarded in order to provide accurate information to the knowledge workers.
4. Concerning privacy, KAs underlie a role-based access control mechanism [Sandhu et al., 96], which, in our case, is primarily based on the permissions within the document repositories integrated into the APOSDLE platform. Therefore, the access permissions have to be resolved from these repository systems, e.g. by utilising mapping techniques.
5. Another requirement of the APOSDLE approach deals with the type and granularity of resources to be described by metadata attributes. Precisely, APOSDLE has to support multimedia content objects, but also virtual objects or parts of documents. Further, the definition of relations to other KAs is necessary.
6. Finally, M3KAR should also tie up to interoperability aspects, i.e. by being capable of supporting existing standards in the field of digital object repositories or even allowing the export of digital objects together with a standard-based description.

Overall, these aspects derived from the Knowledge Artefact lifecycle lead to four dimensions of metadata sets relevant for our purposes: (1) system-managed (read-only) vs. user/application-managed, (2) searchable vs. non-indexed ones, (3) altering vs. constant ones, and (4) temporary vs. persistent ones. In any case, one achievement of the APOSDLE approach is to automatically generate metadata – no matter of what

category. Thus, system-managed attributes, like the URI or permissions, should only be created and updated by M3KAR. Other attributes might automatically be set and modified by other modules or tools.

## 2.2 Functionality and Architecture of the Knowledge Artefact Repository

Beside the depicted aspects of Knowledge Artefacts, the focus of our approach addresses the architectural design as well as the technical realisation of M3KAR. The component itself represents the data layer of the APOSDLE platform, as visualised in Figure 2. Yet, structural and semantic information is provided by other modules, i.e. by the Semantic Repository Manager, the Classification Service or the Semantic Service. The primary task of the Knowledge Artefact Repository comprises the management of Knowledge Artefacts and their provision to the other components of the APOSDLE platform.

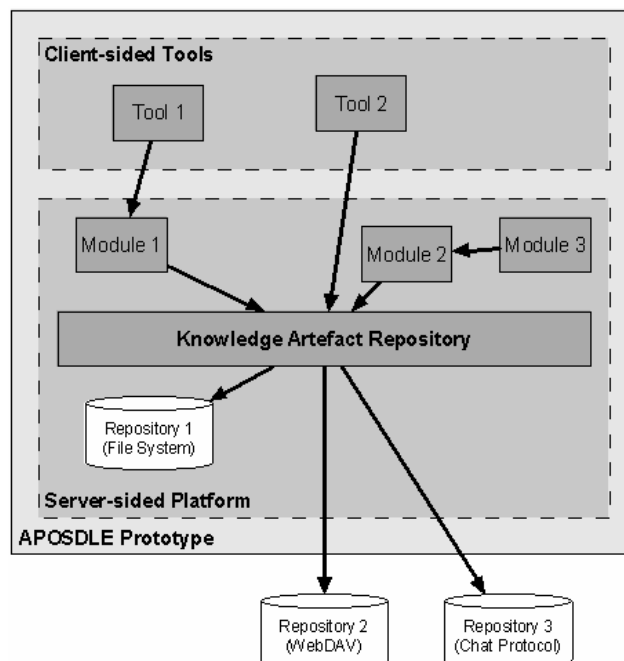


Figure 2: The Concept of a Knowledge Artefact Repository

Figure 2 shows the basic functionality of the M3KAR idea, which deals with the following requirements:

1. A so-called Repository Manager manages the repositories which are integrated into M3KAR. These repositories might be located in or outside the platform, whereby different types of repositories (File System, WebDAV, Chat Protocol) are supported. Technologically, the Product Factory design pattern [Metsker and Wake, 06] is applied within the Repository Manager. As a first step, all necessary operations on a repository

are defined by an interface class. For each repository type, an own implementation of this interface is required. Finally, the repositories can be initialised by a configuration file, but also instantiated on-the-fly without restarting the platform.

2. The repository implementations also support the management of documents on-the-fly, so that documents can be inserted via the platform and, subsequently, new Knowledge Artefacts are made available within the APOSDLE platform. Moreover, also the modification and deletion of KAs are allowed, as already stated in the last subsection. Metadata as well as content imported by other components need to be stored within an own, internal data repository, i.e. a database management system.
3. Due to these first two issues, a notification service is required, so that other modules are informed about changes of and within the repositories. For instance, the Classification Service should be notified in order to initialise the knowledge extraction process and guarantee an accurate index for high-level retrieval functions.
4. Addressing data integration, we face the problem that a digital entity (like a document) can be retrieved from different repositories. As a first and efficient approach to this problem, we propose the calculation of a checksum for digital artefacts and compare their contents, if the checksum is equal. A more sophisticated solution would require semantic comparison of the content in order to detect different versions of artefacts.
5. The logging of the events within M3KAR is important for different reasons, e.g. for examining the usage of KAs in some application scenario or for improving the performance of the overall platform.
6. Finally, performance issues might be of interest, not only for M3KAR, but also for the overall APOSDLE system. Thus, statistical or probabilistic methods [Przybylski, 90], e.g. on basis of access statistics logged within M3KAR, could be utilised to cache the content of KAs which have been very often requested or are predictably of relevance.

Each single aspect is well documented in literature and, further, a lot of systemic types exist in this context. Yet, none solution exactly fulfils the requirements given by the APOSDLE research project, as highlighted in the upcoming section.

### **3 Overview and Limitations of existing Solution Approaches**

Amongst a broad range of solution approaches, relevant types of data layer approaches are inspected in order to outline the differences to the M3KAR idea.

The first category of solutions to be mentioned in this context comprises database management systems itself, which, evidently, lack of the object model for providing operations to other components. Thus, all methods required within the APOSDLE platform must be mapped to a relational database model, which might cost performance and complicates the realisation of other concepts, like caching or session-based metadata. Hence, database technology is necessary for M3KAR no matter whether to integrate a company's explicit knowledge or to manage data objects internally, but it is not sufficient for realising the data layer of APOSDLE.

Secondly, middleware approaches and GRID environments focus on the management of digital objects and their description with metadata, as stated by [Stockinger et al., 01] or with respect to concrete examples like Dinopolis [Haselwanter, 03]. Particularly, the Dinopolis Object supports the issues concerning the KA-lifecycle, while this middleware system is not fully compliant to the server-sided technologies utilised in the APOSDLE project, namely the Spring Framework. In context of M3KAR, an object persistence layer is advantageous and covers many of our requirements on a Knowledge Artefact Repository. Therefore and due to the excellent technological compatibility to the Spring Framework, we decided to apply the Hibernate object persistence layer.

Thirdly and on a higher abstraction level, document and content management systems (and other kinds of knowledge management systems) include such data layer approaches. For instance, the Hyperwave Information Server [Hyperwave, 07] supplies features to manage digital objects and describe them with metadata. Moreover, it is also possible to integrate external data sources by indexing their contents with the full-text indexing engine. Thus, it is possible to make external resources accessible. Furthermore, the Hyperwave Information Server also provides an internal Java-based API to create and manage such Knowledge Artefacts in the way depicted in the last section. However, the commercial character and the functional overheads of such a solution are strong arguments against the utilisation of full-feature products in this scope. Even open source solutions, if customisable to our needs, can not be considered as applicable for the data layer of APOSDLE.

Finally, two research approaches related to M3KAR have to be outlined here: One the one hand, [Kebbell and Campbell, 04] report about a framework which is capable of managing digital objects and their metadata. Although the overall idea is rather similar to M3KAR, it is realised for a small area of a digital library infrastructure, including only one (internal) object repository and not being made available for other projects. On the other hand, [Eriksson and Bang, 06] present a document repository based on semantic documents. Again, this approach addresses only one repository and neglects problems like data integration.

In conclusion of the afore-mentioned, the necessity for realising the idea of the multimedia, multi-source, metadata-based knowledge artefact repository is absolutely given within the APOSDLE research project.

## **4 Current Implementation of M3KAR and Experiences Gained**

At this stage, some aspects of M3KAR have been implemented as a part of the first APOSDLE prototype. This initial version of the M3KAR has been named "Homogeneous Access" (HA) and already realises some of the aspects described in section 2. Yet, this first version of M3KAR still is restricted in many ways:

- Currently, the creation of KAs is not supported. Thus, the KA lifecycle only comprises the retrieval of KA from existing repositories on basis of the URI and the adding of semantic information by means of metadata. System-driven attributes are created on the first access, while all kinds of metadata can be modified. The persisting of changes is not possible, as the first APOSDLE prototype only represents a demonstrator. Consequently, it is not

necessary to deal with the Singleton pattern. KAs are created as multi-state objects, whereby each module uses an own instance of each KA.

- Moreover, HA neither supports KAs on different granularity levels nor multimedia artefacts. Access control on the basis of the permissions from the repositories is also not implemented so far, while access to Knowledge Artefacts is always granted, if a repository is included into the APOSDLE platform. Standard-compliance was non-issue for the first prototype at all.
- The first version of M3KAR already allows the integration of multiple repositories, which was the main objective of Homogeneous Access. At this stage, three types of repositories are supported: (1) file system, (2) WebDAV, and (3) Chat protocols of the Concert Chat Server (the collaboration tool of APOSDLE).
- All repositories have to be configured before start-up of the platform in order to utilise their content. Adding new repositories is not yet supported. The contents of the repositories are being captured once at the system start-up, as a module of the APOSDLE platform, namely the Classification Service, browses through all repositories and indexes their documents. From this moment on, the KAs can be accessed by other components.
- Logging is done to the debug console for debugging reasons only. An own log mechanism for further analysis of the usage of KAs and its exploitation for other purposes is not realised.
- Finally, performance is not considered at all, although it would be an important issue – not only for M3KAR, but also for the overall usability of the APOSDLE system. Concrete performance tests indicated that it took only about one second to retrieve 1000 KAs without content, while it lasted up to 3.5 minutes to deliver a KA including a document with a size of about 1MB thousand times in a row. In this context, it is recommended to apply at least statistical caching methods to improve the systemic performance for multiple deliveries of contents.

As shown with this first version of M3KAR, some of the concepts depicted in section 2 are already realised, others might be of interest or even necessary for the next APOSDLE prototype. Particularly, the persistence of KAs and further aspects of their lifecycle as well as caching approaches are required to guarantee an efficient, usable APOSDLE system.

## **5 Conclusions and Future Work**

In summarising this paper, we have to state that our idea of a “Multimedia, Multi-source, Metadata-based Knowledge Artefact Repository” represents not only a data-layer for APOSDLE but also comprises an advantageous approach to integrate and semantically enrich a company’s explicit knowledge which might be distributed over several repositories. Moreover, M3KAR is a key component for a usable APOSDLE platform. By now, we have implemented a first version of M3KAR, examined it and identified necessary or even critical issues to be considered within the next APOSDLE prototype. On that account, we have already started to plan and implement these new features.

## Acknowledgements

APOSDLE is partially funded by the FP6 of the European Commission within the IST work programme 2004 (FP6-IST-2004-027023).

## References

- [APOSDLE, 07] APOSDLE, “learn@work”, Project Website, 2007, <http://www.aposdle.tugraz.at/> (2007-05-16).
- [Borghoff and Pareschi, 97] U.M. Borghoff, and R. Pareschi, “Information Technology for Knowledge Management”, *Journal of Universal Computer Science*, 3(8), 1997, pp. 835-842.
- [Dzbor et al., 00] M. Dzbor, J. Paralic, and M. Paralic, “Knowledge Management in a distributed organization”, Technical Report KMI-TR-94, Knowledge Media Institute, Open University, 2000.
- [Eriksson and Bang, 06] H. Eriksson, and M. Bang, “Towards Document Repositories Based On Semantic Documents”, *Proceedings of International Conference on Knowledge Management*, 2006, pp. 313-320.
- [Haselwanter, 03] E. Haselwanter, „Aspects of component composition in distributed frameworks”, master’s thesis, Graz University of Technology, 2003.
- [Hyperwave, 07] Hyperwave, “Company’s Website”, 2007, <http://www.hyperwave.com/e/> (2007-06-16).
- [Kebbell and Campbell, 04] A. Kebbell, and D. Campbell, “Managing digital objects and their metadata: challenges and responses”, *Proceedings of International Conference on Dublin Core and Metadata Applications*, 2004.
- [Kühn and Abecker, 97] O. Kühn, and A. Abecker, “Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges”, *Journal of Universal Computer Science*, 3(8), 1997, pp. 929-954.
- [Ley, 06] T. Ley, “APOSDLE Glossary: Workplace Learning, Work-integrated Learning”, Know Center, 2006, [http://www.aposdle.tugraz.at/weblog/aposdle\\_glossary\\_workplace\\_learning\\_work\\_integrated\\_learning](http://www.aposdle.tugraz.at/weblog/aposdle_glossary_workplace_learning_work_integrated_learning) (2007-05-16).
- [Metsker and Wake, 06] S.J. Metsker, and W.C. Wake, “Design Patterns in Java”, 2<sup>nd</sup> edition, Boston: Pearson, 2006.
- [Pant and Ondo, 02] Y. Pant, and K. Ondo, “Thread Specific Singleton: Handling singleton pattern errors in multi-threaded applications and their variations”, *Journal of Object Technology*, 1(2), 2002, pp. 155-169.
- [Przybylski, 90] S.A. Przybylski, “Cache and Memory Hierarchy Design: A Performance Directed Approach”, San Francisco: Morgan Kaufmann Publishers, 1990.
- [Sandhu et al., 96] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, „Role-Based Access Control Models”, *Computer*, 29(2), 1996, pp. 38-47.
- [Stockinger et al., 01] H. Stockinger, O.F. Rana, R. Moore, and A. Merzky, “Data Management for Grid Environments”, *Proceedings of International Conference on High-Performance Computing and Networking*, 2001, pp. 151-160.