Knowledge Resource Object Development and Mining with a Knowledge-centric Architecture

Claus-Peter Rückemann
Westfälische Wilhelms-Universität Münster (WWU), Germany;
Knowledge in Motion, DIMF, Germany;
Leibniz Universität Hannover, Germany
Email: ruckema@uni-muenster.de

***Abstract*—This paper presents a knowledge resources' view on object development based on the new architecture framework, which is a research result of a series of practical problem solving implementations and further developments based on common knowledge concepts and integrated application components. The framework is considered knowledge-centric, supported by knowledge resources, which constitute the fundamental base and imply the core of key assets. Besides further knowledge development, the knowledge-centric architecture flexibly allows implementations of mining and computation components for many scenarios and the employment of available computation infrastructures. An important quality of the architecture framework is the inherent property to assign different roles for the professional tasks in creation and development cycles. These roles address the major complements of knowledge, including factual, conceptual, and procedural components as well as documentation. This paper discusses object development and knowledge mining based on a knowledge-centric architecture. The main goal of this research is to illustrate the universal multi-disciplinary and multi-lingual knowledge and content related features for major use-case activities, deploying a knowledge-centric architecture and to provide new examples from the practice of systematical object development and the implementation of flexible methods.**

***Keywords*–Knowledge Resource Object Creation; Object Development and Integration; Containers and Collections; Knowledge-centric Architecture; Knowledge-centric Mining.**

I. INTRODUCTION

From information science view, knowledge-centric complements are in focus when creating and employing knowledge resources for mining purposes. Creation and further development of these resources and objects is of primary interest. This extended research is based on the knowledge-centric computation architecture and its deployment for knowledge mining, which was presented at the INFOCOMP 2019 conference in Nice, France [1]. The presented research concentrated on the new architecture and facilities to implement and describe activity groups in a consistent and standardised way.

Consequently, this extended paper concentrates on research from the knowledge resources' perspective, the object development, based on major use-cases groups, namely knowledge creation and development on the one hand and knowledge mining on the other hand. The development of knowledge resources and objects essentially contributes to the long-term values created for universal knowledge. Fostering structured organisation of content, which can be addressed by manifold methods and which can carry universal references to multi-disciplinary and multi-lingual knowledge is considered a primary benefit for development, valorisation, and flexibility of re-use.

The continuous development of knowledge resources and objects should be provided a number of major features, which are supported by the architecture, especially:

- Development of resources and objects,
- flexible object structures,
- object groups, e.g., collections and containers,
- multi-disciplinary knowledge object support,
- flexible object and context references,
- support of knowledge complements,
- multi-lingual features,
- consistency of resources,
- object re-use,
- knowledge mining,
- creation and deployment of algorithms,
- object integration,
- a fundamental, universal framework with a
- high potential of long-term facilities.

The goal of this research is to implement resources and features according to the knowledge-centric architecture. Use-cases illustrate and summarise the knowledge and content related features for major activities and to summarise the architecture and the implementation fundaments. The paper introduces major use-cases of object development and knowledge mining based on the architecture and presents relevant results for objects and content. The resulting object development based on this research is presented and demonstrated for a consistent view on objects, closely referring to activities for these use-cases.

This paper is organised as follows. Section II and III deliver motivation, fundaments and previous work. Section III presents the architecture resulting from this research. Sections IV and V introduce to the knowledge-centric architecture and to an implementation of two major use-cases for object development and mining. Section VI presents a resulting object development architecture. Sections VII and VIII present the respective results of object development and knowledge mining, illustrated by practical examples. Section IX discusses the main results and evaluates them in context of the practical application scenario. Section X summarises the results and lessons learned, conclusions, and future work.

II. MOTIVATION

All implementations of mathematical machines, which we call 'computer systems' today, can strictly only deal with formal systems. Knowledge is a capability of a living organism and can itself not be incorporated by formal systems. Neither can intrinsic meaning, which is an essential characteristics of real knowledge and a unique stronghold of knowledge, be a matter of formal systems nor can mathematical relations, the theory of sets, exlusiveness or creating completeness be applied to knowledge.

Solutions requiring a wide range of knowledge content as well as implementations of algorithms and components are often challenging to handle, the more when it comes to operating and adding to the resulting solutions for decades or even further developing content and implementations for long-term. Over time, the further developments and services are becoming more complicated without a common, holistic frame for content and implementation. When gathering a large number of independent implementations, we experienced an increasing heterogeneity in content development but also in implementations of computing components.

This background is the major motivation for the development of an advanced framework based on long-term Knowledge Resources and integrated application components providing a valuable means of tackling the challenges. Nevertheless, in complex cases even major component groups cannot protect long-term challenges, if there is no basic framework architecture enabling to care for knowledge and computational implementation. The practice of creating solutions, which have to deal with the complements of knowledge suggests that flexible but nevertheless methodological, systematical approaches are required. The goal of this research is to create such knowledge-centric architecture, based on a wide range of multi-disciplinary implementations and practical case studies in different disciplines and dealing with different foci, for many years. While further developing and updating the knowledge related attributes, data, implementations, and solutions, all of them had to be revisited over time, improving and where necessary recreating implementation and developing content.

Knowledge resources and originary resources cover the complements of factual, conceptual, procedural, and metacognitive complements, e.g., from collections and referenced resources. The architecture presented here aims to seamlessly integrating separate roles of contributing parties, e.g., scientific staff creating research data, professional classification by experienced research library specialists, and developers of application components as well as services. The guideline was enabling to retain the knowledge required to resemble the inherent complexity of realia situations, real and material instead of abstract situations, while allowing lex parsimoniae principles of William of Ockham for problem solving.

Overall outcome and new insight gained from the practical knowledge resource object development and mining solutions based on the designed and implemented knowledge-centric architecture are presented in the following sections.

III. Fundaments and Previous Work

With one of the best and most solid works, Aristotle outlined the fundaments of terminology and of understanding knowledge [2] being an essential part of 'Ethics' [3]. Information sciences can very much benefit from Aristotle's fundaments and a knowledge-centric approach, e.g., by Anderson and Krathwohl [4], but for building holistic and sustainable solutions they need to go beyond the available technology-based approaches and hypothesis [5] as analysed in Platons' Phaidon. So far, there is no other practical advanced knowledge-centric architectural specification known, which implements these fundaments. Making a distinction and creating interfaces between methods and applications [6], the principles are based on the methodology of knowledge mapping [7]. The implementation can make use of objects and conceptual knowledge [8] and shows being able to build a base for application scenarios like associative processing [9] and advanced knowledge discovery [10]. Based on this background, during the last decades, a number of different case solutions were created, implemented, and realised on this fundament, including: Dynamical visualisation, knowledge mining, knowledge mapping, Content Factor, phonetic algorithms, Geoscientific Information Systems (GIS), Environmental Information Systems (EIS), cartographic mapping, service design, service management, and High End Computation. All such implementations include extensive use of LX Knowledge Resources, as explicitly representing the developed resources in this notation, and computation algorithms. This paper, presenting the new architecture, does not allow to illustrate too many implementation details. Therefore, an excerpt of practical solutions is cited, which have been reimplemented by the collaboration of the participated research groups and published, creating a base for this architecture. Representative examples are a) integrated systems and supercomputing resources used with phonetic algorithms and pattern matching [11] for knowledge mining [12], b) multi-dimensional context creation based on the methodology of Knowledge Mapping [13], and c) an exemplary resulting, widely used conceptual knowledge subset for geo-spatial scenarios [14]. The LX Knowledge Resources cover the factual, conceptual, procedural, and metacognitive complements in all cases, e.g., from collections and referenced resources.

An understanding of the essence and complexity of universal, multi-disciplinary knowledge can be achieved by taking a closer look on classification. The state-of-the-art of classifying 'universal knowledge' is the Universal Decimal Classification (UDC) [15] and its solid background, flexibility, and long history. The LX Knowledge Resources' structure and the classification references [16] based on UDC [17], [18] are essential means for the processing workflows and evaluation. Both provide strong multi-disciplinary and multi-lingual support. For the research, all small unsorted excerpts of the Knowledge Resources objects only refer to main UDC-based classes, which for this publication are taken from the Multilingual Universal Decimal Classification Summary (UDCC Publication No. 088) [17] released by the UDC Consortium under the Creative Commons Attribution Share Alike 3.0 license [19] (first release 2009, subsequent update 2012). These components and their qualities are integrated in the resulting architecture with the methodologies and systematic use.

IV. Knowledge-centric Architecture

As discussed above, the presented results were achieved, based on the knowledge-centric architecture created for a series of previously implemented problem solutions and Knowledge Resources developments over the last years.

A. General Computation Architecture

The complements diagram of the implementation architecture [20][21][22] is shown in Figure 1. The major components are core resources and module resources. The result resources include object collections, which result from the application of core and module resources in arbitrary scenarios.

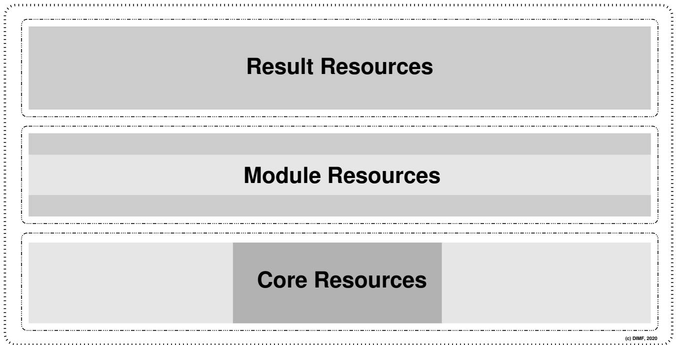

Figure 1. Complements diagram of the resources components architecture, including the three main complements of core, module, and result resources.

The sizes of this figure and the associated complements diagrams correspond, the following figures show complementary details from this context.

The core resources in this architecture comprise required resources. The complements diagram (Figure 2) shows the essential detail.

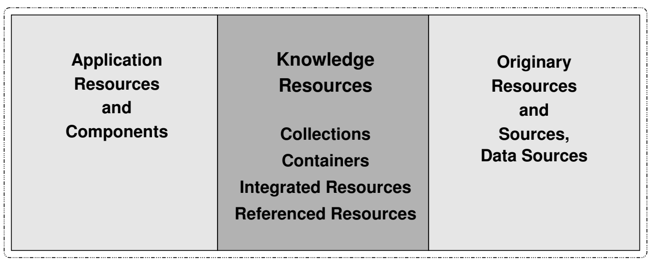

Figure 2. Computation architecture: Complements diagram of general core resources, from originary to knowledge and application resources.

The core resources can be divided into three categories: The central LX Knowledge Resources, originary resources, and application resources and components. The first, the Knowledge Resources, can include collections and containers as well as integrated resources and references to resources. The second, the originary resources, can include realia and original sources, which in many cases may have instances in the Knowledge Resources. The third, the application resources, can include implementations of algorithms, workflows, and procedures, which form applications and components. Instances of these components can also be employable in solutions due to their procedural nature, e.g., in module resources.

The complements diagram of general module resources is shown in Figure 3.

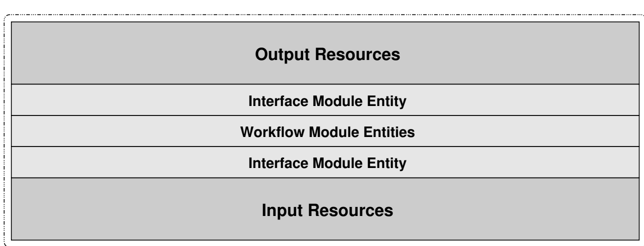

Figure 3. Computation architecture: Complements diagram of module resources, from input, interfaces and workflow entities to output.

A general set of module resources consists of input resources, modules, and output resources. The central workflow module entities are accompanied by interface module entities for input and output resources. For many architecture implementations, chains of module resources can be created, which can, for example, be used in pipeline and in parallel.

B. Architecture Complements for Knowledge Mining

For the case of knowledge mining, the complements diagram of the core resources is shown in Figure 4.

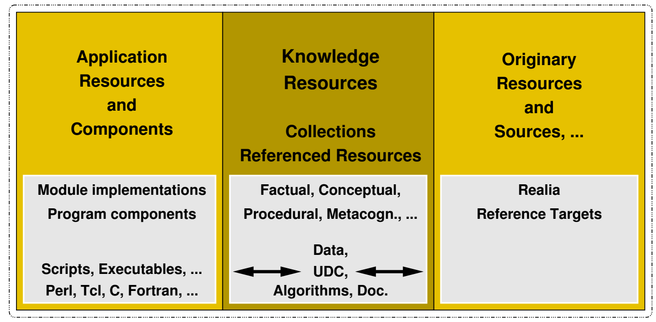

Figure 4. Knowledge Mining: Complements diagram of the core resources and examples of their contribution implementations.

Application resources and components are based on module implementations and program components for the knowledge mining realisation. Implementations employ scripting, high level languages, and third party components. Knowledge Resources and originary resources cover the complements of factual, conceptual, procedural, and metacognitive complements, e.g., from collections and references resources.

The respective complements diagram of a module resource for a text based knowledge mining implementation consists of several features (Figure 5).

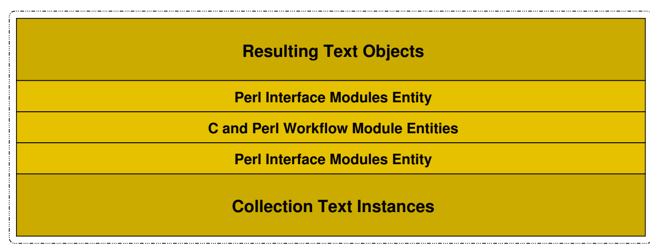

Figure 5. Knowledge Mining: Complements diagram of a module resource used for creating module chains for text based knowledge mining.

The input and output resources consist of text object instances in text based cases of knowledge mining. Here, the module entity implementations were implemented in C and Perl [23] for the respective implementations. The interface module entities are implemented in Perl, with the option to be on-the-fly generated within a workflow.

The knowledge-centric architecture does focus on resources and application scenarios, one of the most important is computation cases. Large computation workflow chains can be built with the architecture as was demonstrated with the reimplemented solutions for different cases, which were above referred to. An implementation sequence of module resources can be considered an intermediate step in building a workflow. Results within an implementation sequence can be considered intermediate results and instances, e.g., from the integrated mining of collection and container resources.

V. Object Development and Mining Cases

The goal was to create a knowledge-centric computation architecture, which allows a close integration of Knowledge Resources with wide spectra of complementary knowledge and flexible, efficient computational solutions, while being able to specify practically required roles for creation and long-term development. The knowledge-centric architecture can provide a base for an arbitrary range of use-cases and associated components. Two major groups of use-cases are

- resources creation and development and
- knowledge mining and selected associated methods.

A. Major Use-case Groups

Figure 6 shows an use-case diagram (UML, Unified Modeling Language) of the knowledge-centric computation architecture.

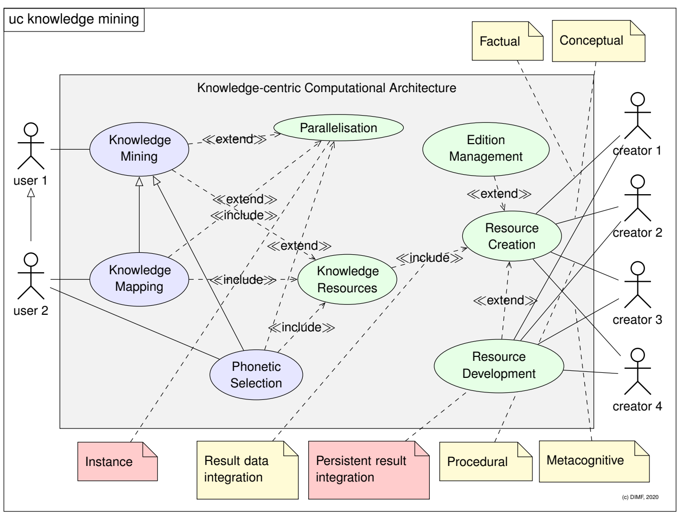

Figure 6. Use-case diagram of the knowledge-centric computation architecture: Two major use-case groups with four creator roles.

The excerpt illustrates an integrated view on the two groups of knowledge mining (bluish), which was implemented spanning knowledge mining use-cases and knowledge creation and development use-cases (greenish). In this widely deployed scenario, the implementation does have two main types of actors, namely creators and users. The use-cases have different actors, two 'user' roles and four 'creator' roles.

The selected system context is given by the grey box. The selected use-cases (ellipses) can be distinguished in use-cases for creators (greenish: resource creation, resource development, edition management, parallelisation) and use-cases for users (bluish: knowledge mining, knowledge mapping, phonetic selection).

Knowledge mining is supported by and using the cases of knowledge mapping and phonetic selection, which inherit to the knowledge mining instance the implemented methods and algorithms contributed by other user groups. For clearness, the creator and other roles for these two cases are not included in this diagram. Knowledge mining, mapping, and phonetic selection include the use-case of LX Knowledge Resources. The cases of this group are extended by parallelisation, respective computation, here instance based workflow parallelisation, which enables the computation-relevant optimisation for individual implementations and infrastructures.

The knowledge resources include the use-case of resource creation, which allows to integrate persistently added results. The use-case of resource creation itself is extended by the use-cases of resource development and edition management, which allows to define editions of resources for consistency in advanced complex application scenarios.

The use-case scenario reflects the professional practice of having separate roles for creating and developing factual, conceptual, procedural, and documentation, respective metacognitive knowledge. In most cases, different specialists are employed for creating and developing

- factual knowledge, e.g., research data and its documentation,
- conceptual knowledge, e.g., classification of knowledge objects,
- procedural knowledge, e.g., procedures, workflows, programs, and their respective documentation,
- metacognitive knowledge, e.g., documentation of experiences.

In practice, the creators are commonly represented by different groups, e.g., scientists, classification experts in scientific libraries, and designers of scientific algorithms and workflows.

B. Main Components

The core components of a basic knowledge mining implementation with the LX Knowledge Resources, based on the knowledge-centric computation architecture, can be summarised with a block diagram (Figure 7).

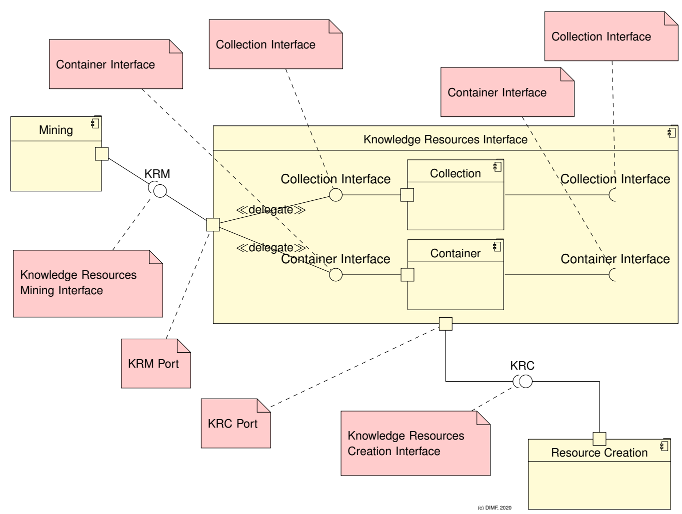

Figure 7. Block diagram of respective knowledge-centric computation architecture components: LX Knowledge Resources and interfaces excerpt.

The block diagram shows the LX Knowledge Resources and two types of knowledge object groups, namely object collections and containers. Each type and implementation can have individual and specialised interfaces. Knowledge mining is provided by an interface with the LX Knowledge Resources. The diagram also contains the interface block, due to the importance of the resource creation use-case. The individual groups have ports, interaction points, which can be used via interfaces, e.g., Knowledge Resources Creation (KRC) port and Knowledge Resources Mining (KRM) port. Components can have further interfaces, with and without delegating ports. It is common that independent resources are in many cases not necessarily orchestrated.

C. Activity Groups

A number of activities are associated with different components. An important activity regarding the resource creation is the creation-update (Figure 8).

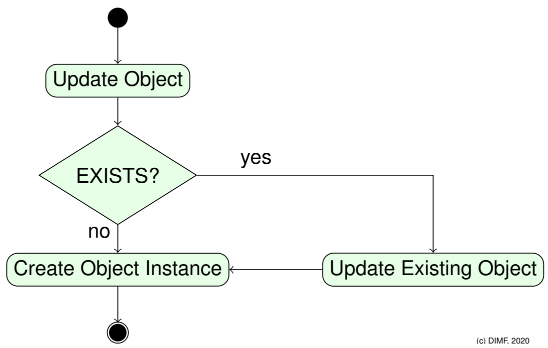

Figure 8. Activity diagram illustrating the essential object creation-update activities in the knowledge-centric computation architecture.

The resource creation component has to provide creation and update activities for the different creator groups. A simple but important example for resource creation and development is the creation of an object instance and respective updating an existing object with a new instance.

Start state is any state of the knowledge resources. End state is a new state of the knowledge resources. Regarding resource creation and development use-cases the start and end states should be considered intermediate states. As shown in the use-case diagram, professional practice affords the implementation of according activities for all required, specialised creator roles. A fundamental mining activity with knowledge resources is a resource request targeting to create an intermediate result (Figure 9).

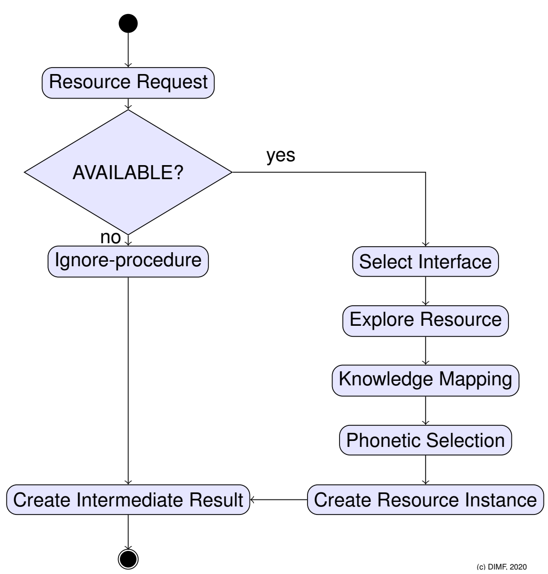

Figure 9. Activity diagram of a basic resource request for creating an intermediate knowledge mining result.

Start state is any state in a knowledge mining workflow chain. End state is a new state in a knowledge mining workflow chain. Regarding knowledge mining use-cases the start and end states should be considered intermediate states. If a resource is not available then an ignore-procedure continues for creating an intermediate result. The ignore-procedure can contribute its status to the workflow chain. If the resource is available then an interface is selected for the resource exploration. The exploration can use available activities, e.g., knowledge mapping and phonetic selection, in order to create a resource instance, which contributed to creating an intermediate result for the workflow chain.

Examples of activities are multi-dimensional context creation by knowledge mapping [7] and phonetic association, e.g., using Soundex [24][25][26]. Sample Soundex codes developed [12] are used for names in various textual, contextual, and linguistical situations, implemented in order to be integrated in a large number of situations.

VI. Resulting Object Development Architecture

Any sustainable long-term knowledge resources require an efficient concept for continuous content development. The fundamental concept for continuous content development implemented and practiced with this long-term research is shown in a Knowledge Resource Object Development (KROD) diagram (Figure 10).

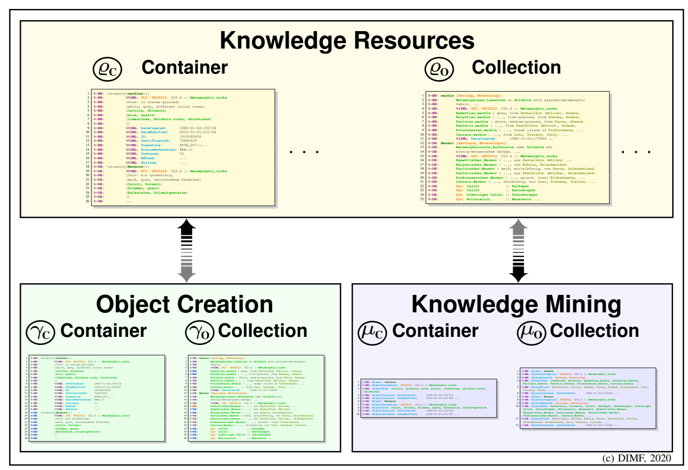

Figure 10. Knowledge Resource Object Development Diagram: Knowledge Resources, Object Creation, Knowledge Mining, containers and collections.

The diagram displays formalised activities associated with use-cases and their relations from the view of real object instances. The diagram illustrates activity groups and respective excerpts of objects.

Two major activities in context of LX Knowledge Resources (yellowish) are the creation of objects (greenish) and knowledge mining (bluish). Here, some major object groups are shown, collections and containers. The respective object instances (ϱ, γ, and μ groups) and properties are shown and further discussed in separate object instance representation views in the next sections. Different colours are used, which correspond to the discussed use-cases (Figures 6, 7, 8, and 9).

Table I shows the legend of colours/symbols used with the development components: The case groups (Knowledge Resources, KR; Object Creation, OC; Knowledge Mining, KM) and objects.

Table I. Legend for development components: Case groups (Knowledge Resources, KR; Object Creation, OC; Knowledge Mining, KM) and objects.

Development Component	*Colour Name*	*Colour*
Case group KR	(yellowish)	
Case group OC	(greenish)	
Case group KM	(bluish)	
Case group KR object	(darker yellowish)	
Case group OC object	(darker greenish)	
Case group KM object	(darker bluish)	
Objects, object terms	dark green	
Object terms' context	light green	
Object places' context	olive green	
Object keywords	orange	
Object instance data	cyan	
Object development-is	violet	
Object development-in	blue	
Object development-out	red	

The major groups of objects, container (C) objects and collection (O) objects [27], correspond with instances when used: Knowledge resources' objects in containers and collections, ϱ_{C} and ϱ_{O}, Object Creation's objects of containers and collections, γ_{C} and γ_{O}, and Knowledge Mining's objects from containers and collections, μ_{C} and μ_{O}.

A creation algorithm $\mathcal{C}$ realises a function $f_{\mathcal{C}}$ out of the set of input data γ and the set of output data ϱ, this is $f_{\mathcal{C}} : \gamma \rightarrow \varrho$.

A mining algorithm $\mathcal{M}$ realises a function $f_{\mathcal{M}}$ out of the set of input data ϱ and the set of output data μ, this is $f_{\mathcal{M}} : \varrho \rightarrow \mu$.

For the major C and O groups of objects, the algorithms for these two operations result in two groups of functions:

$$f_{\mathcal{C}} : \gamma_{\text{C,O}} \longrightarrow \varrho_{\text{C,O}} \tag{1}$$

$$f_{\mathcal{M}} : \varrho_{\text{C,O}} \longrightarrow \mu_{\text{C,O}} \tag{2}$$

Therefore, this architecture provides any algorithm a flexibility to be implemented by an arbitrary number of implementations, e.g., manual, automated, and hybrid approaches. Even more important is the development of the objects covering the complements of factual, conceptual, procedural, and metacognitive knowledge. Development policies and enforcement of policies are matter of implementing operation and services based on workflows and algorithms, which are beyond demonstration here. The following sections present the use-cases and object development, which are in focus of this paper.

VII. Object Development Case Results

The three types of instances of respective C and O objects, which are in focus of this research, are discussed in the following sections.

Excerpts of the LX Knowledge Resources provided for this research are shown using automatic component highlighting according to the above legend. Line-wise modification indicators were used for illustration in this paper.

A. Resources and Objects

The objects in the LX Knowledge Resources can contain content and documentation of factual, conceptual, procedural, and metacognitive knowledge. There are multi-fold means of attainment for this goal.

The selected resources contain more than 100,000 associated objects, entities, and references, especially referring to a potential of more than 4,000 minerals and mineral synonyms as well as possible 3,000,000 reference objects and conceptual knowledge in more than 50 languages.

The LX Knowledge Resources case group (yellowish), according to the KROD Diagram (Figure 10), is represented by excerpts illustrating object development examples from geoscientific container and collection objects. Figure 11 shows a small excerpt of a ϱ_{C} container object representation.

```
%-EN: \stoentry{marbles}{}
%-EN:           %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%-EN:           {fine- to coarse-grained}
%-EN:           {white, grey, different colour tones}
%-EN:           {calcite, dolomite}
%-EN:           {mica, quartz}
%-EN:           {limestones, dolomite rocks, dolostones}
%-EN:           {}
%-EN:           %%IML: DateCreated:       1989-01-02:192718
%-EN:           %%IML: DateModified:      2020-01-01:231704
%-EN:           %%IML: ID:                1910093458
%-EN:           %%IML: CertificateID:     75660429
%-EN:           %%IML: Signature:         DF98_007::...
%-EN:           %%IML: SourceReferences: #KR::C
%-EN:           %%IML: Instance:          52
%-EN:           %%IML: Md5sum:            ...
%-EN:           %%IML: Sha1sum:           ...
%-DE: \stoentry{Marmore}{}
%-DE:           %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%-DE:           {fein- bis grobkörnig}
%-DE:           {weiß, grau, verschiedene Farbtöne}
%-DE:           {Calcit, Dolomit}
%-DE:           {Glimmer, Quarz}
%-DE:           {Kalksteine, Dolomitgesteine}
%-DE:           {}
%-DE:           ...
```

Figure 11. Object instance representation: LX Knowledge Resources KR, container object (excerpt).

The auto-highlighting emphasises various knowledge in context of this object instance. The object entity of type stone is 'marbles' (dark green). Material components (light green) are calcite and dolomite (major components), mica and quartz (minor components). Educts are limestone, dolomite rocks etc. Conceptual knowledge is available from the Universal Classified Classification (UCC) concordances [10] as UDC2012 (orange), classifying the object with a reference to the major group `UDC:552.4`, metamorphic rocks. Here, e.g., the conceptual knowledge refers to metamorphic rocks. The object carries multi-lingual instances, English and German (indicated by `%-EN:` and `%-DE:`). This excerpt is instance number `52`. Here, the shown excerpts indicate that instances of this object entity were created and developed over significant date range. Figure 12 shows a small excerpt of a ϱ_{O} collection object representation.

```
%-EN: marble [Geology, Mineralogy]:
%-EN:         Metamorphised_limestone or dolomite with grained-metamorphic
%-EN:         fabric. ...
%-EN:         %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%-EN:         Hymettian_marble : grey, from Pentelikon (Attica), Greece.
%-EN:         Karystian_marble : ..., fine-grained, from Euboea, Greece.
%-EN:         Parisian_marble : white, medium-grained, from Paros, Greece.
%-EN:         Pentelic_marble : ..., from Pentelikon (Attica), Greece.
%-EN:         Proconnesian_marble : ..., Greek island of Prokonnesos, ...
%-EN:         Carrara_marble : ..., from Luni, Tuscany, Italy. ...
%-EN:         %%IML: DateCreated:      1988-12-26:175694 ...
%-DE: Marmor [Geologie, Mineralogie]:
%-DE:         Metamorphisierte_Kalksteine oder Dolomite mit
%-DE:         körnig-metamorphem Gefüge. ...
%-DE:         %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%-DE:         Hymettischer_Marmor : ..., aus Pentelikon (Attika) ...
%-DE:         Karystischer_Marmor : ..., von Euboia, Griechenland.
%-DE:         Parisischer_Marmor : weiß, mittelkörnig, von Paros, Griechenland.
%-DE:         Pentelischer_Marmor : ..., aus Pentelikon (Attika), Griechenland.
%-DE:         Prokonnesischer_Marmor : ..., griech. Insel Prokonnesos, ...
%-DE:         Carrara_Marmor : ..., feinkörnig, aus Luni, Toscana, Italien. ...
%-DE:         Syn: Calcit             :: Kalkspat
%-DE:         Syn: Calcit             :: Kanonenspat
%-DE:         Syn: blättriger Calcit :: Schieferspat
%-DE:         Syn: Nitrocalcit        :: Mauersalz ...
```

Figure 12. Object instance representation: LX Knowledge Resources KR, collection object (excerpt).

The LX Knowledge Resources' collection object contains a 'marble' and 'Marmor' entity (dark green), material components (light green), keywords ('Geology', 'Mineralogy'), conceptual knowledge from Universal Classified Classification (UCC) concordance references as UDC2012 (orange), classifying the object with a reference to the major group `UDC:552.4`, metamorphic rocks.

The representations and highlighting also show the lingual contexts (violet), e.g., English, dates regarding the object instance and so on. The collection object carries multi-lingual instances, English and German (indicated by `%-EN:` and `%-DE:`).

The object instances carry their instance data (cyan). Instance representation, e.g., identification, is available from the object and entity instances, Object Envelopes (OEN), [28]. A larger excerpt of the instance data of the ϱ_{O} object is shown in Figure 13.

```
%-EN:        %%IML: DateCreated:       1988-12-26:175694
%-EN:        %%IML: DateModified:      2020-01-01:232927
%-EN:        %%IML: ID:                1909042178
%-EN:        %%IML: CertificateID:     74581231
%-EN:        %%IML: Signature:         DF98_007::...
%-EN:        %%IML: SourceReferences:  #KR::O
%-EN:        %%IML: Instance:          119
%-EN:        %%IML: Md5sum:            ...
%-EN:        %%IML: Sha1sum:           ...
```

Figure 13. Object instance representation: Instance data excerpt of LX Knowledge Resources collection object (Figure 12).

The excerpt shows creation and modification dates, ids, references, and check sums.

B. Object Creation and Development

The object creation case group (greenish), according to the KROD Diagram (Figure 10), is represented by following the developments of object examples presented in the above resources section. Figure 14 shows a small excerpt of a γ_{C} container object representation.

```
%-EN: \stoentry{marbles}{}
%-EN:          %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%-EN:          {fine- to coarse-grained}
%-EN:          {white, grey, different colour tones}
%-EN:          {calcite, dolomite}
%-EN:          {mica, quartz}
%-EN:          {limestones, dolomite rocks, dolostones}
%-EN:          {}
%-EN:          %%IML: DateCreated:       1989-01-02:192718
%-EN:          %%IML: DateModified:      2020-01-01:225807
%-EN:          %%IML: ID:                1910093458
%-EN:          %%IML: CertificateID:     75660429
%-EN:          %%IML: Signature:         DF98_007::...
%-EN:          %%IML: SourceReferences:  #KR::C
%-EN:          %%IML: Instance:          51
%-EN:          %%IML: Md5sum:            ...
%-EN:          %%IML: Sha1sum:           ...
%>DE: \stoentry{Marmore}{}
%>DE:          %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%>DE:          {fein- bis grobkörnig}
%>DE:          {weiß, grau, verschiedene Farbtöne}
%>DE:          {Calcit, Dolomit}
%>DE:          {Glimmer, Quarz}
%>DE:          {Kalksteine, Dolomitgesteine}
%>DE:          {}
%>DE:          ...
```

Figure 14. Object instance representation: Object Creation OC container object (excerpt).

The OC object illustrates the object creation and development process, an update of an existing object on base of a previously created object instance in the Knowledge Resources. This excerpt is based on instance number `51`, creating instance number `52` (Figure 11). The excerpt shows the addition of a language entry to the container. Here, an entity in German language is added to the object. In this case, the added material is indicated by line-based blue markers, `%>DE:`. Figure 15 shows a small excerpt of a γ_{O} collection object representation.

```
%-EN: marble [Geology, Mineralogy]:
%-EN:        Metamorphised_limestone or dolomite with grained-metamorphic
%-EN:        fabric. ...
%-EN:        %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%>EN:        Hymettian_marble : grey, from Pentelikon (Attica), Greece.
%>EN:        Karystian_marble : ..., fine-grained, from Euboea, Greece.
%-EN:        Parisian_marble : white, medium-grained, from Paros, Greece.
%-EN:        Pentelic_marble : ..., from Pentelikon (Attica), Greece.
%-EN:        Proconnesian_marble : ..., Greek island of Prokonnesos, ...
%-EN:        Carrara_marble : ..., from Luni, Tuscany, Italy. ...
%-EN:        %%IML: DateCreated:      1988-12-26:175694 ...
%-DE: Marmor [Geologie, Mineralogie]:
%-DE:        Metamorphisierte_Kalksteine oder Dolomite mit
%-DE:        körnig-metamorphem Gefüge. ...
%-DE:        %%IML: UCC: UDC2012: 552.4 :: Metamorphic_rocks
%<DE:        Pentelische_Marmore : ..., aus Pentelikon (Attika) ...
%>DE:        Hymettischer_Marmor : ..., aus Pentelikon (Attika) ...
%>DE:        Karystischer_Marmor : ..., von Euboia, Griechenland.
%-DE:        Parisischer_Marmor : weiß, mittelkörnig, von Paros, Griechenland.
%-DE:        Pentelischer_Marmor : ..., aus Pentelikon (Attika), Griechenland.
%-DE:        Prokonnesischer_Marmor : ..., griech. Insel Prokonnesos, ...
%-DE:        Carrara_Marmor : ..., feinkörnig, aus Luni, Toscana, Italien. ...
%>DE:        Syn: Calcit             :: Kalkspat
%>DE:        Syn: Calcit             :: Kanonenspat
%>DE:        Syn: blättriger Calcit :: Schieferspat
%>DE:        Syn: Nitrocalcit        :: Mauersalz ...
```

Figure 15. Object instance representation: Object Creation OC collection object (excerpt).

The excerpt shows modification of entries in a collection object. Entries are added (`%>EN:` and `%>DE:`, blue colour) and removed (`%<DE:`, red colour). The object creation adds new types of rock in two languages and a number of synonyms, which are unique to German language. A larger excerpt of the instance data of the γ_{O} object is shown in Figure 16.

```
%-EN:        %%IML: DateCreated:       1988-12-26:175694
%<EN:        %%IML: DateModified:      2017-05-30:221532
%>EN:        %%IML: DateModified:      2020-01-01:231704
%-EN:        %%IML: ID:                1909042178
%-EN:        %%IML: CertificateID:     74581231
%-EN:        %%IML: Signature:         DF98_007::...
%-EN:        %%IML: SourceReferences:  #KR::O
%<EN:        %%IML: Instance:          118
%>EN:        %%IML: Instance:          119
%<EN:        %%IML: Md5sum:            ...
%>EN:        %%IML: Md5sum:            ...
%<EN:        %%IML: Sha1sum:           ...
%>EN:        %%IML: Sha1sum:           ...
```

Figure 16. Object instance representation: Instance data excerpt of Object Creation collection object (Figure 15).

The excerpt in state of development contains references of instances `118` and `119` of the ϱ_{O} object.

VIII. Knowledge Mining Case Results

A. Objects and mining

The Knowledge Mining case group (bluish), according to the KROD Diagram (Figure 10), is represented by following the developments of object examples presented in the above resources and creation/development sections. Figure 17 shows a small excerpt of a μ_{C} container object extract representation.

```
%-EN: Object: marbles
%-EN: ObjectConceptual: UDC2012: 552.4 :: Metamorphic_rocks
%-EN: ObjectTerm: calcite, dolomite, mica, quartz, limestones, dolomite rocks,
 dolostones
%-EN: ObjectInstance: DateCreated:       1989-01-02:192718
%-EN: ObjectInstance: DateModified:      2020-01-01:231704 ...
%-DE: Object: Marmore
%-DE: ObjectConceptual: UDC2012: 552.4 :: Metamorphic_rocks
%-DE: ObjectTerm: Calcit, Dolomit, Glimmer, Quarz, Kalksteine, Dolomitgesteine
%-DE: ObjectInstance: DateCreated:       1989-01-02:192718
%-DE: ObjectInstance: DateModified:      2020-01-01:231704 ...
```

Figure 17. Object instance representation: Knowledge Mining KM container object extract (excerpt).

Figure 18 shows a small excerpt of an extracted μ_{O} collection object representation.

```
%-EN: Object: marble
%-EN: ObjectConceptual: UDC2012: 552.4 :: Metamorphic_rocks
%-EN: ObjectKeyword: Geology, Mineralogy
%-EN: ObjectTerm: limestone, dolomite, Hymettian_marble, Karystian_marble,
 Parisian_marble, Pentelic_marble, Proconnesian_marble, Carrara_marble, ...
%-EN: ObjectPlace: Pentelikon, Attica, Euboea, Paros, Greece, Prokonnesos, Luni,
  Tuscany, Italy. ...
%-EN: ObjectInstance: DateCreated:      1988-12-26:175694 ...
%-DE: Object: Marmor
%-DE: ObjectConceptual: UDC2012: 552.4 :: Metamorphic_rocks
%-DE: ObjectKeyword: Geologie, Mineralogie
%-DE: ObjectTerm: Kalksteine, Dolomite, Calcit, Kalkspat, Kanonenspat,
 blättriger Calcit, Schieferspat, Nitrocalcit, Mauersalz, Hymettischer_Marmor,
 Karystischer_Marmor, Parisischer_Marmor, Pentelischer_Marmor,
 Prokonnesischer_Marmor, Carrara_Marmor, ...
%-DE: ObjectPlace: Pentelikon, Attika, Euboia, Paros, Pentelikon, Attika,
 Griechenland, Prokonnesos, Luni, Toscana, Italien, ...
%-DE: ObjectInstance: DateCreated:      1988-12-26:175694 ...
```

Figure 18. Object instance representation: Knowledge Mining KM collection object extract (excerpt).

A larger excerpt of the instance data of the extract μ_{O} object is shown in Figure 19.

```
%-EN:        %%IML: DateCreated:       1988-12-26:175694
%-EN:        %%IML: DateModified:      2020-01-01:232927
%-EN:        %%IML: ID:                1909042178
%-EN:        %%IML: CertificateID:     74581231
%-EN:        %%IML: Signature:         DF98_007::...
%-EN:        %%IML: SourceReferences:  #KR::O
%-EN:        %%IML: Instance:          119 [extract]
%-EN:        %%IML: Md5sum:            ...
%-EN:        %%IML: Sha1sum:           ...
```

Figure 19. Object instance representation: Instance data excerpt of Knowledge Mining collection object extract (Figure 18).

The purpose of intermediate results in a universal context is to provide a high flexibility and modularity in workflows. It must be possible to create and re-use algorithms, which are taking care for arbitrary scenarios. An integrated intermediate result of such case, a resource request on 'marble', creating an intermediate result including object instances from μ_{O} and μ_{C}, is displayed in Figure 20.

```
%-EN: Object: marble, marbles
%-EN: ObjectConceptual: UDC2012: 552.4 :: Metamorphic_rocks
%-EN: ObjectKeyword: Geology, Mineralogy
%-EN: ObjectTerm: limestone, dolomite, calcite, mica, quartz, limestones,
 dolomite rocks, dolostones, Hymettian_marble, Karystian_marble, Parisian_marble
 , Pentelic_marble, Proconnesian_marble, Carrara_marble, ...
%-EN: ObjectPlace: Pentelikon, Attica, Euboea, Paros, Greece, Prokonnesos, Luni,
  Tuscany, Italy, ...
%-EN: ObjectInstance: SourceReferences: #KR::C::1910093458::52 #
 KR::O::1909042178::119
%-DE: Object: Marmor, Marmore
%-DE: ObjectConceptual: UDC2012: 552.4 :: Metamorphic_rocks
%-DE: ObjectKeyword: Geologie, Mineralogie
%-DE: ObjectTerm: Kalksteine, Dolomite, Dolomit, Calcit, Glimmer, Quarz,
 Kalksteine, Dolomitgesteine, Kalkspat, Kanonenspat, blättriger Calcit,
 Schieferspat, Nitrocalcit, Mauersalz, Hymettischer_Marmor, Karystischer_Marmor,
  Parisischer_Marmor, Pentelischer_Marmor, Prokonnesischer_Marmor,
 Carrara_Marmor, ...
%-DE: ObjectPlace: Pentelikon, Attika, Euboia, Paros, Griechenland, Prokonnesos,
  Luni, Toscana, Italien, ...
%-DE: ObjectInstance: SourceReferences: #KR::C::1910093458::52 #
 KR::O::1909042178::119
```

Figure 20. Object instance representation: Knowledge Mining intermediate result from objects' extracts (excerpt).

This excerpt is only one of many possible, different intermediate results. The algorithms of creating intermediate results allow arbitrary integration and formalisation, abstraction, reduction. Any available methods can contribute, including mining algorithms, knowledge mapping, and phonetic selection. This example of an integrated intermediate result contains structured information from multi-lingual object entities, material, and location information, from both collection and container objects. The intermediate result resulting from the mining process is integrating knowledge from both multi-lingual collection and container instances (`#KR::C::1910093458::52` and `#KR::O::1909042178::119`), as referring to the previously shown instances `52` of the ϱ_{C} object and `119` of the ϱ_{O} object. Here, the excerpt shows references from a stones container and a geoscientific collection. Further width and depth of integrated knowledge in this context can include any multi-disciplinary references, e.g., references can range from mineralogy containers to geoscientific synonym containers.

IX. Discussion

The continuous object development and knowledge mining cases results were efficiently and effectively supported deploying the features provided by the object development architecture and functions.

- Development of resources and objects were successfully implemented and demonstrated for object groups in collections and containers.
- Object structures have shown to allow highest flexibility. All components support multi-disciplinary knowledge objects. All groups and objects allow flexible object and context references, support of knowledge complements, e.g., conceptual knowledge via UDC, and support multi-lingual content and access.
- All components can ensure consistency of resources, object re-use, advanced knowledge mining, creation and deployment of algorithms, and object integration.
- The fundamental, universal framework is already successfully developed and used in practice for many years before publication.

The knowledge-centric architecture proved being a flexible, reliable, and robust fundament for object creation and development as well as for advanced knowledge mining, managing and using LX Knowledge Resources objects' collections and containers for many years. The architecture provides the flexibility that workflow chain modules and whole workflow chains can be employed sequential or parallel. The components in general are not limited by the architecture to be implemented for synchronous or asynchronous accesses if required for arbitrary algorithms and workflows.

The implementations for practical case studies built upon this architecture span different disciplines and deal with different foci. The excerpted cases include general, simplified cases of knowledge mining and practical knowledge development scenarios from realisations, which were implemented for large practical solutions. These cases are relevant because of the professional background and practice required to deal with development of resources and application components for long-term tasks.

In complex scenarios, different disciplines contribute fulfilling different tasks. In case of knowledge creation and development and its valorisation different specialised expertise is required. In general, content and applications are created by different disciplines. Even different aspects of content may require different specialists groups, different roles, e.g., natural sciences research data and conceptual valorisation are often done by scientists from a respective discipline and information scientists. Many components have to be revisited and improved over time as the results and facilities should be continued and preserved and be available for long-term. In the implementation cases, factual, conceptual, procedural, and metacognitive knowledge is cared for by different experts. The architecture allows flexible and efficient separation of roles. For example, research data can be created by a role and can at any time be amended with classification and procedural documentation by experienced research library specialist and researcher roles.

The LX Knowledge Resources are containing a lot more content and references than can be used at present time in most cases. The architecture allowed to support retaining the associated knowledge required to resemble the intrinsic complexity of realia situations while implementing selected solutions for isolated as well as complex situations. The development of knowledge mining and the provisioning of services based on these tasks can continuously be done by application developers, accessing the continuously extendable LX Knowledge Resources.

X. Conclusion

This paper presented a knowledge resources' view on object development based on a new knowledge-centric architecture. The paper presented the architecture and discussed two successfully implemented major use-cases, object development and knowledge mining. The implementation supports all the features, which were targeted and should be provided regarded the architecture and object development.

Knowledge resource object development and respective object instance representation for resources, object creation and development, and knowledge mining were presented for different types of object groups, namely collections and containers.

The paper presented consistent view on activities and real object instances from use-case scenarios in context with the research results of creating a knowledge-centric computation architecture. The resulting architecture was developed in continuous cross development of multi-disciplinary, multilingual knowledge resources and practical knowledge-centric solutions. This paper presented the major qualities of the computation architecture. The practical employment of the architecture was illustrated for advanced knowledge mining and practical development uses-cases. The contributing research collaboration achieved to create a practical approach for a knowledge-centric computation architecture, which allows the methodological and systematical development and employment of components, including knowledge resources. The architecture covers the creation of flexible solutions, which allow to most widely employ the complements of knowledge.

The long-term knowledge resource scenarios and the knowledge-centric architecture and use-cases proved in practice to support the object development, the seamless separation and integration of roles for different disciplines and tasks while implementing and realising solutions based on knowledge resources and computation. In addition to the implemented and referred case studies, it was shown that object development and major use-cases can be efficiently managed. Especially, on the one hand, knowledge creation and development can be professionally dealt with by groups from responsible disciplines. On the other hand, knowledge mining can be based on the work of these disciplines while service based use and implementation can be given different roles, relying on the resources being continuously in development.

Future research will concentrate on further extending and developing knowledge resources in order to foster the creation of content bases and to provide long-term capacities and creating new advanced algorithms and mining workflows for enabling fundaments for new insight, participating different institutions and roles, based on the knowledge-centric computation architecture and knowledge resources.

Acknowledgment

We are grateful to the "Knowledge in Motion" (KiM) long-term project, Unabhängiges Deutsches Institut für Multidisziplinäre Forschung (DIMF), for partially funding implementations, case studies, publications, and for granting access to LX Knowledge Resources under grants D2016F5P04648, D2017F1P04812, and D2018F6P04938 and to its senior scientific members and members of the permanent commission of the science council and the board of trustees, especially to Dr. Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek (GWLB) Hannover, to Dipl.-Biol. Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, to Dipl.-Ing. Martin Hofmeister, Hannover, and to Olaf Lau, Hannover, Germany, for collaboration, practical multi-disciplinary case studies, and the analysis and implementation of advanced concepts. We are grateful to Dipl.-Ing. Hans-Günther Müller, HPE, Germany, for his excellent contributions and assistance, providing practical private cloud and storage solutions. We are grateful to all national and international partners in the Geo Exploration and Information cooperations for their constructive and transdisciplinary support. We are grateful to the Science and High Performance Supercomputing Centre (SHPSC) for long-term support and The International ARS Science and History Network for providing multi-disciplinary application scenarios and assistance. / DIMF-PIID-DF98_007.

References

[1] C.-P. Rückemann, "A Knowledge-centric Computation Architecture and the Case of Knowledge Mining," in Proceedings of The Ninth International Conference on Advanced Communications and Computation (INFOCOMP 2019), July 28 – August 1, 2019, Nice, France. XPS Press, Wilmington, Delaware, USA, 2019, pp. 35–40, ISSN: 2308-3484, ISBN: 978-1-61208-732-0, URL: http://www.thinkmind.org/index.php?view=article&articleid=infocomp_2019_3_20_60044 [accessed: 2020-05-02].

[2] Aristotle, Nicomachean Ethics, 2008, (Written 350 B.C.E.), Translated by W. D. Ross, Provided by The Internet Classics Archive, URL: http://classics.mit.edu/Aristotle/nicomachaen.html [accessed: 2020-05-02].

[3] Aristotle, The Ethics of Aristotle, 2005, Project Gutenberg, eBook, EBook-No.: 8438, Release Date: July, 2005, Digitised Version of the Original Publication, Produced by Ted Garvin, David Widger, and the DP Team, Edition 10, URL: http://www.gutenberg.org/ebooks/8438 [accessed: 2020-05-02].

[4] L. W. Anderson and D. R. Krathwohl, Eds., A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon, Boston, MA (Pearson Education Group), USA, 2001, ISBN-13: 978-0801319037.

[5] Plato, Phaedo, 2008, (Written 360 B.C.E.), Translated by Benjamin Jowett, Provided by The Internet Classics Archive, URL: http://classics.mit.edu/Plato/phaedo.html [accessed: 2020-05-02].

[6] C.-P. Rückemann and F. Hülsmann, "Significant Differences: Methodologies and Applications," KiMrise, Knowledge in Motion Meeting, November 27, 2017, Knowledge in Motion, Hannover, Germany, 2017.

[7] C.-P. Rückemann, "Methodology of Knowledge Mapping for Arbitrary Objects and Entities: Knowledge Mining and Spatial Representations – Objects in Multi-dimensional Context," in Proceedings of The Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2018), March 25 – 29, 2018, Rome, Italy. XPS Press, Wilmington, Delaware, USA, 2018, pp. 40–45, ISSN: 2308-393X, ISBN-13: 978-1-61208-617-0, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2018_3_20_30078 [accessed: 2020-05-02].

[8] C.-P. Rückemann, "Creation of Objects and Concordances for Knowledge Processing and Advanced Computing," in Proceedings of The Fifth International Conference on Advanced Communications and Computation (INFOCOMP 2015), June 21–26, 2015, Brussels, Belgium. XPS Press, 2015, ISSN: 2308-3484, ISBN-13: 978-1-61208-416-9, URL: http://www.thinkmind.org/index.php?view=article&articleid=infocomp_2015_4_30_60038 [accessed: 2020-05-02].

[9] C.-P. Rückemann, "Advanced Association Processing and Computation Facilities for Geoscientific and Archaeological Knowledge Resources Components," in Proceedings of The Eighth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2016), April 24 – 28, 2016, Venice, Italy. XPS Press, 2016, pp. 69–75, ISSN: 2308-393X, ISBN-13: 978-1-61208-469-5, URL: http://www.thinkmind.org/download.php?articleid=geoprocessing_2016_4_20_30144 [accessed: 2020-05-02].

[10] C.-P. Rückemann, "Advanced Knowledge Discovery and Computing based on Knowledge Resources, Concordances, and Classification," International Journal On Advances in Intelligent Systems, vol. 9, no. 1&2, 2016, pp. 27–40, ISSN: 1942-2679, URL: http://www.iariajournals.org/intelligent_systems/intsys_v9_n12_2016_paged.pdf [accessed: 2020-05-02].

[11] "Perl Compatible Regular Expressions (PCRE)," 2020, URL: https://www.pcre.org/ [accessed: 2020-05-02].

[12] C.-P. Rückemann, "Archaeological and Geoscientific Objects used with Integrated Systems and Scientific Supercomputing Resources," International Journal on Advances in Systems and Measurements, vol. 6, no. 1&2, 2013, pp. 200–213, ISSN: 1942-261x, URL: http://www.iariajournals.org/systems_and_measurements/sysmea_v6_n12_2013_paged.pdf [accessed: 2020-05-02].

[13] C.-P. Rückemann, "Multi-dimensional Context Creation Based on the Methodology of Knowledge Mapping," International Journal on Advances in Software, vol. 11, no. 3&4, 2018, pp. 286–298, ISSN: 1942-2628, URL: http://www.iariajournals.org/software/soft_v11_n34_2018_paged.pdf [accessed: 2020-05-02].

[14] C.-P. Rückemann, "Superordinate Knowledge Based Comprehensive Subset of Conceptual Knowledge for Practical Geo-spatial Application Scenarios," in Proceedings of The Eleventh International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2019), February 24 – 28, 2019, Athens, Greece. XPS Press, Wilmington, Delaware, USA, 2019, pp. 52–58, ISSN: 2308-393X, ISBN: 978-1-61208-687-3, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2019_3_30_30039 [accessed: 2020-05-02].

[15] UDC, Universal Decimal Classification. British Standards Institute (BSI), 2005, complete Edition, ISBN: 0-580-45482-7, Vol. 1 and 2.

[16] C.-P. Rückemann, "Enabling Dynamical Use of Integrated Systems and Scientific Supercomputing Resources for Archaeological Information Systems," in Proceedings of the International Conference on Advanced Communications and Computation (INFOCOMP 2012), October 21–26, 2012, Venice, Italy, 2012, pp. 36–41, ISBN: 978-1-61208-226-4, URL: http://www.thinkmind.org/download.php?articleid=infocomp_2012_3_10_10012 [accessed: 2020-05-02].

[17] "Multilingual Universal Decimal Classification Summary," 2012, UDC Consortium, 2012, Web resource, v. 1.1. The Hague: UDC Consortium (UDCC Publication No. 088), URL: http://www.udcc.org/udcsummary/php/index.php [accessed: 2020-05-02].

[18] "UDC Online," 2020, URL: http://www.udc-hub.com [accessed: 2020-05-02].

[19] "Creative Commons Attribution Share Alike 3.0 license," 2012, URL: http://creativecommons.org/licenses/by-sa/3.0/ [accessed: 2020-05-02].

[20] B. Oestereich and A. Scheithauer, Analyse und Design mit der UML 2.5: Objektorientierte Softwareentwicklung (in English: Analysis and Design With the UML 2.5: Object-oriented Software Development), 11th ed. De Gruyter Oldenbourg, 2013, ISBN: 978-3486721409.

[21] "Unified Modeling Language (UML)," 2020, URL: http://uml.org/ [accessed: 2020-05-02].

[22] "About the Unified Modeling Language Specification 2.5.1," Dec. 2017, OMG – Object Management Group, URL: https://www.omg.org/spec/UML/ [accessed: 2020-05-02].

[23] "The Perl Programming Language," 2020, URL: https://www.perl.org/ [accessed: 2020-05-02].

[24] R. C. Russel and M. K. Odell, "U.S. patent 1261167," 1918, (Soundex algorithm), patent issued 1918-04-02.

[25] D. E. Knuth, The Art of Computer Programming: Sorting and Searching. Addison-Wesley, 1973, vol. 3, ISBN: 978-0-201-03803-3, OCLC: 39472999, pp. 391–392.

[26] National Archives and Records Administration (NARA), "The Soundex Indexing System," 2007, 2007-05-30, URL: http://www.archives.gov/research/census/soundex.html [accessed: 2020-05-02].

[27] C.-P. Rückemann, "Integration of Knowledge Resources and Advanced Association Processing for Geosciences and Archaeology," International Journal on Advances in Systems and Measurements, vol. 9, no. 3&4, 2016, pp. 485–495 ISSN: 1942-261x, URL: http://www.iariajournals.org/systems_and_measurements/sysmea_v9_n34_2016_paged.pdf [accessed: 2020-05-02].

[28] C.-P. Rückemann, "High End Computing Using Advanced Archaeology and Geoscience Objects," International Journal On Advances in Intelligent Systems, vol. 6, no. 3&4, 2013, ISSN: 1942-2679, LCCN: 2008212456, URL: http://www.iariajournals.org/intelligent_systems/intsys_v6_n34_2013_paged.pdf [accessed: 2020-05-02].