Creating Knowledge-based Dynamical Visualisation and Computation

Claus-Peter Rückemann
Westfälische Wilhelms-Universität Münster (WWU),
Leibniz Universität Hannover,
North-German Supercomputing Alliance (HLRN), Germany
Email: ruckema@uni-muenster.de

***Abstract*—The research conducted in this paper presents an implementation for creating knowledge-based dynamical visualisation and computation. The core knowledge is based on long-term knowledge resources further developed for several decades and used with many applications scenarios utilising multi-disciplinary and multi-lingual content and context like references, associations, and knowledge container collections. A major goal of the application case studies shown here is creating context-sensitive dynamical program components and algorithms from selected knowledge content. The selections are results of dynamical workflows, which are part of component implementations, e.g., including search processes and result matrix generators. Previous research has shown that long-term knowledge resources are the most important and most valuable component of long-term approaches and solutions. Here, the structures and classifications are used with independent database frameworks and programming interfaces. The results show that the methodological foundations and knowledge resources are very well suitable as long-term core base, as well as for creating dynamical application components, e.g., for visualisation and computation in multi-disciplinary, geoscientific, and spatial context. The knowledge resources can refer to any kind of resources. The overall environment allows to develop and govern extensive content structures and promote their long-term vitality.**

***Keywords*–Advanced Knowledge Discovery; Universal Decimal Classification; Conceptual Knowledge; Dynamical Visualisation and Computation; Geoscientific Applications.**

I. INTRODUCTION

Long-term Knowledge resources can be created and used for universal documentation and re-use of content. The re-use includes discovery, as well as gathering new results and creating new applications. The knowledge resources [1] can refer to any kind of resources, e.g., to natural sciences resources or historical geographic resources [2]. Basics of knowledge organisation [3] and multi-lingual lexical linked data [4] have been discussed for various disciplines and shown the huge potential and value of the knowledge. This also shows the benefits of linking with universal classifications, especially with consequently numerical notations, which can be easily and most flexibly and efficiently used with modern applications components. Further, on the one hand, information services benefit from a comprehensive and holistic model for evaluation [5] and on the other hand, they align with the benefits for a quality management of information services [6].

The paper presents a new implementation for creating knowledge-based dynamical visualisation and computation, which have not been integrated before for that purpose. Therefore, a major goal of the application case studies is creating dynamical program components and algorithms based on knowledge resources. The different previous projects and case studies have already shown that the combination of knowledge resources with integrated conceptual knowledge references can be used for the creation of dynamical applications.

The dynamical visualisation and computation based on knowledge resources does have numberless applications. Some prominent examples with the research presented here are knowledge discovery, visualising result matrices from workflows or search processes, and creating objects and extending knowledge resources. The framework presented here is a high level framework interconnecting several frameworks for complex system architecture, multi-column operation, and long-term creation for main resources. Therefore, the required approach is considered to be necessarily most complex from knowledge and implementation point of view.

Following the Geo Exploration and Information case studies [7] based on the actmap framework [8] a number of developments for the deployment of High End Computing resources and technologies with integrated systems are still state of the art. In addition, including the structural and conceptual knowledge based on the knowledge resources, research has been done for a different special database framework, which is as well autonomous and can be used for the creation of standalone dynamical and portable application components. The components can be integrated with the existing frameworks, as well as they can be used as standalone interactive applications.

This research shows details of the latest case studies and discusses the up-to-date experiences from the implementation of the dynamical components and their integration with knowledge resources' structures and workflows.

This paper is organised as follows. Section II introduces the methodological bases, Section III presents the fundamental implementations based on knowledge resources and computation, discussing the foundations, architecture, framework, integration, and dynamical visualisation and computation. Section IV shows the details of the implementation and the resulting components, from geo sets, computation to index selection and some views from the resulting visualisations. Sections V and VI evaluate the main results and summarise the lessons learned, conclusions and future work.

II. STATE-OF-THE-ART AND MOTIVATION

The creation of long-term knowledge resources and utilisation methods is one of the most pressing goals in information science as the masses of data and the loss of knowledge in society are steadily increasing in all areas. Existing projects employ segment-like spectra of disciplines in their focus. Examples are large digital libraries and projects like the Europeana [9] and the World Digital Library (WDL) [10].

As existing projects, e.g., which are only concentrating on bibliographic means, do not focus on such integration, use different and mostly isolated classifications and schemes for different areas and specific purposes. For example, there is a small number of general classifications, which are mostly used in library context. Although such classifications are used in many thousands of institutions worldwide this is neither a general use case or application scenario nor a significant share of the overall knowledge. They are missing to provide facilities for arbitrary kinds of objects, e.g., factual data and trans-disciplinary context in information science and natural sciences.

In contrast to that, the state of the art for documentation of universal, conceptual knowledge is the Universal Decimal Classification (UDC) [11], which is one of the very few classifications providing a universal classification. Besides public interfaces, the implementations of the known application scenarios are not publicly available in common. Anyhow, all known scenarios have in common that they deploy only a small subset of available classifications and in the vast majority the classification process is not automated. It is necessary to develop logical structures in order to govern the existing big data today and in future, especially in volume, variability, and velocity. This is necessary in order to keep the information addressable and maintain the quality of data on long-term.

Beyond the focus of the mentioned segment-like projects the knowledge resources and concepts discussed and implemented in this research focus on the trans-disciplinary integration of arbitrary different segments and disciplines and a universal usability based on factual data and criteria. The documentation and context also integrates and refers to content and context, e.g., conceptual, procedural, and metacognitive knowledge and allows for a huge range of possible scenarios. This is a driving force to extend the use of classification in trans- and multi-disciplinary context and transfer the experiences from deploying a classification for long-term documentation and application.

III. Methodological bases: Logical structures

To work on that goal requires to define information units and to care for depositing an appropriate segmentation in sub-units. The information units require links to the related units, e.g., superunits. The challenges are to define these structures and units for data used in different disciplinary context, in one discipline, as well as in multi-disciplinary context. These logical structures are the basic precondition for the development of functioning algorithms, which can access the units and whose application can be perfected in a next self-learning step. The tries of using unstructured data result in the fact that data volumes, variabilities, and volumes devaluate the resulting values of requests. Any isolated technological approaches to the big data challenge have shown not to be constructive. A sustainable approach has to consider the data and structure itself.

- The first step is the preconditional definition of a logical, commonly valid structure for the data.
- The second step is the planning for the applications based on the logical structures in step one.
- The third step is the creation of algorithms regarding the data and data retrieval, interfaces, and workflows based on steps one and two. The fourth step is the planning of the implementation. This includes data format, platforms, and applications.

Further, the creation, development, and operation of the content and components require to consider and define the essential plans, especially:

- Plans for extending structures.
- Preparations for all required interfaces for the newly extended structures should be done.
- Plans for self-learning components.
- Plans for container formats and utilisation.
- Plans for sustainable handling of data lifecycles, data formats, and system resources,

The early stage of planning requires a concept catalogue. So far, the activities are commonly in a pre-planning stage. The next step includes case studies on structures, algorithms, and collaborational issues (efficiency, economical cycle).

IV. Implementation: Knowledge and computation

The implementation shows the context-sensitive dynamical components based on the knowledge resources. The knowledge resources provide the structure and integrate the factual knowledge, the references, including the references for the object classification views required for the dynamic utilisation, ensuring integration [12] and sustainability [13].

Previous case studies have shown that any suitable cartographic visualisation can be used for the presentation of the results, for example, with the Generic Mapping Tools [14] (e.g., filtering, trend fitting, gridding, views, and projections) or creating exports and imports with various products. Most available cartographic visualisation products are too specialised in order to handle advanced knowledge workflows one the one hand and dynamical results on cognitive context on the other hand. In the presented case where the application should concentrate on the intention of presenting a special result in an abstract way we require special and flexible facilities for dynamical sketch drawings. The more, in the special case the cognitive background forbids to concentrate on detailed cartographic visualisation or mixing with modern ways of geographic conventions. Historical names, locations, and context are not adequately represented by existing modern frameworks.

Regarding both requirements for this study are fully complied by the flexibility of the implementation. The knowledge resources themselves are not restrictive regarding the use of other components for other purposes.

 ISBN: 978-1-61208-383-4

A. Implementation foundations

The implementation for dynamical visualisation and computation is based on the framework for the architecture for documentation and development of advanced scientific computing and multi-disciplinary knowledge [15]. The architecture implemented for an economical long-term strategy is based on different development blocks. Figure 1 shows the three main columns: Application resources, knowledge resources, and originary resources. The central block in the “Collaboration house” framework architecture [16], are the knowledge resources, scientific resources, databases, containers, and documentation (e.g., LX [1], databases, containers, list resources). These can be based on and refer to the originary resources and sources (photos, scientific data, literature). The knowledge resources are used as a universal component for compute and storage workflows. Application resources and components (Active Source, Active Map, local applications) are implementations for analysing, utilising, and processing data and making the information and knowledge accessible. The related information, all data, and algorithm objects presented are copyright the author of this paper, LX Foundation Scientific Resources [1], all rights reserved.

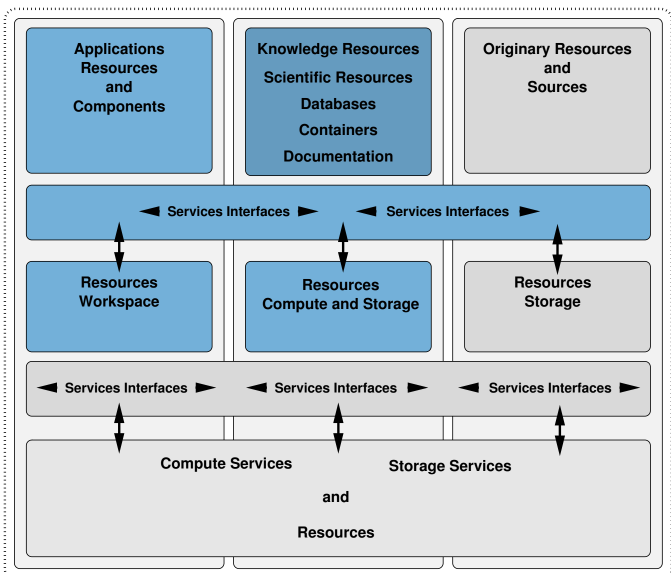

Figure 1. Architecture: Columns of practical dimensions. The knowledge resources are the central component within the long-term architecture.

The LX structure and the classification references based on UDC [11], especially mentioning the well structured editions [17] and the multi-lingual features [18], are essential means for the processing workflows and evaluation of the knowledge objects and containers. Both provide strong multi-disciplinary and multi-lingual support. The three blocks are supported by services’ interfaces. The interfaces interact with the physical resources, in the local workspace, in the compute and storage resources the knowledge resources are situated, and in the storage resources for the originary resources. All of these do allow for advanced scientific computing and data processing, as well as the access of compute and storage resources via services interfaces. The resources’ needs depend on the application scenarios to be implemented for user groups.

B. Implementation and integration

The context of the application components is fully integrated with the knowledge resources and dynamical components [19]. The screenshot (Figure 2) illustrates some features.

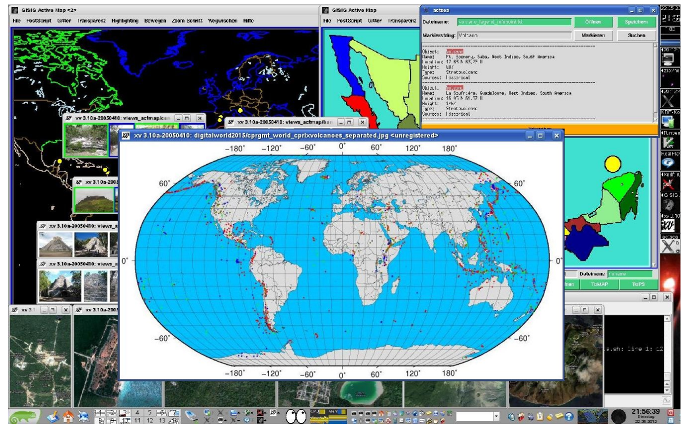

Figure 2. Dynamical use of information systems and scientific computing with multi-disciplinary and universal knowledge resources [16].

Shown examples illustrate features of Active Source, computed and filtered views, LX information, and aerial site photographs, e.g., from Google Maps. Many general aspects of dynamical use of information systems and scientific computing have been analysed with the collaboration house case studies.

C. Implemented content and application dynamics

The main groups of challenges are resulting from the content and from the applications. From content side: The knowledge resources provide the central repository and infrastructure (‘Knowledge as a Service’) for discovery and component creation. From application side: The dynamical components can deploy the resources to any extent and in any step of workflows.

Any part of the components and features can be assembled from the knowledge resources’ workflows. As an example, the site context and factual data components, the database and the graphical user interface components including event definition and management can be dynamically created via transform routines and `concatenate` operations.

D. Dynamical visualisation and computation implementation

A number of different visualisation tools and frameworks have been analysed in the latest case studies. The results presented here were mostly realised with Tcl/Tk [20] for the dynamical visualisations, Fortran and C based programs for required algorithms, Message Passing Interface (MPI) [21], and Perl for dynamical scripting.

Many components have been developed for the actmap framework [8] and successfully used and verified in context with existing scenarios. Besides the actmap framework additional possibilities of creating application programming interfaces and graphical user interfaces for dynamical visualisation of knowledge matrices have to be analysed.

As a simple example for a dynamical, portable, and standalone component an application like Tclworld has been considered [22]. The application is built on a very portable Tool Command Language (TCL) base integrating programming, database, and user interfaces.

The database application programming interface [23] is very simple, portable, and extendable. The database graphical user interface [24] can be used within the same application and is based on a rapid prototyping concept. Both interface models allow dynamical control and extension of any features regarding the application, as well as for the content.

V. Implementation and resulting components

Creating knowledge-based dynamical components starts at the application to knowledge resources' interface level. The conventional components and processes have been described and discussed in practice in detail in previous research, regarding Active Map Software [8], case studies [7], and knowledge integration for classification and computation [12].

For the visualisation and computation using the matrix objects with spatial and georeferenced context, a new application instance "lxworlddynamic" has been created based on the knowledge resources and interfaces. This component re-using the Tclworld interfaces is the required extension complementary to the actmap framework components. All parts of the component shall

- integrate with the knowledge resources and existing components, e.g., georeferenced objects,
- be usable interactively,
- have access to the content, e.g., index selection,
- facilitate a standalone application assembly, and
- allow a flexible configuration of all aspects of the applications' visualisation and computation.

Especially, the component requires a site handler database, has to refer to sets of georeferenced objects, a level handler for selecting levels of detail, to create groups of matrix objects, to support for an individualised configuration, to facilitate index selections on the generated matrix elements, to visualise the matrix elements and context graphically, and has to provide associated data textually and numerically.

A. Knowledge resources and geo sets

The integrated information systems can generate result matrices based on the available components and workflows. The result matrix generators can be configured to deliver any kind of result matrix. One base for the implementation is the generation of georeferences data resulting from requests. The listing (Figure 3) shows an example of a result matrix, an excerpt of the generated site handler database.

```
{LX Site 20.687652 -88.567674} site 20.687652 88.567674
{LX Site 20.682658 -88.570147} site 20.682658 88.570147
{LX Site 20.682859 -88.568548} site 20.682859 88.568548
{LX Site 21.210859 -86.80352} site 21.210859 86.80352
{LX Site 21.097633 -86.796799} site 21.097633 86.796799
{LX Site 21.157199 -86.834736} site 21.157199 86.834736
{LX Site 21.157199 -86.834736} site 21.157199 86.834736
{LX Site 21.094751 -86.812248} site 21.094751 86.812248
{LX Site 41.377968 2.17804} site 41.377968 -2.17804
{LX Site 41.375842 2.177696} site 41.375842 -2.177696
{LX Site 38.676439 -0.198618} site 38.676439 0.198618
{LX Site 38.677683 -0.198103} site 38.677683 0.198103
{LX Site 21.234502 -86.740494} site 21.234502 86.740494
{LX Site 21.184412 -86.807528} site 21.184412 86.807528
{LX Site 16.043421 -61.663857} site 16.043421 61.663857
{LX Site 16.043153 -61.663374} site 16.043153 61.663374
{LX Site 17.633225 -63.236961} site 17.633225 63.236961
{LX Site 17.633225 -63.236961} site 17.633225 63.236961
{LX Site 51.151786 10.415039} site 51.151786 -10.415039
{LX Site 20.214301 -87.429103} site 20.214301 87.429103
{LX Site 20.493276 -87.735701} site 20.493276 87.735701
{LX Site 20.494663 -87.720294} site 20.494663 87.720294
{LX Site 20.494761 -87.720138} site 20.494761 87.720138
{LX Site 40.821961 14.428868} site 40.821961 -14.428868
{LX Site 20.365228 -87.452545} site 20.365228 87.452545
{LX Site 20.365228 -87.452545} site 20.365228 87.452545
```

Figure 3. Excerpt of generated site handler database (lxworlddynamic).

The database is the result of a request summarising results on objects referring to a defined context, in this case references between archaeological and geological objects.

The framework provides a number of features like level handlers and sets of object georeferences. The listing (Figure 4) shows an excerpt of the generated site level handler.

```
foreach i {
  { ... }
  { ... }
  ...
} {+ $i level 2}
```

Figure 4. Excerpt of generated site level handler (lxworlddynamic).

The level handler manages the site handler database, which can also be generated and updated dynamically. Appropriate entries are managed by the geo::set. The listing (Figure 5) shows an excerpt of the generated geo set.

```
geo::Set {
  { ... } site ... ...
  { ... } site ... ...
  ...
}
```

Figure 5. Excerpt of generated geo::set (lxworlddynamic).

Groups of objects, e.g., associated archaeological, geological, meteorite, and volcanological sites, as well as subgroups like pottery and stones, can be dynamically associated and handled in the generated component. The listing (Figure 6) shows an excerpt of the generated database matrix.

```
 + {Archaeological site} : {A selected site with findings
    of human activity, complementary to a
    {Geological site} .
    These sites have been dynamically created
    from a request to the LX knowledge resources ... .
    These sites have been ... .
    }

 + {Geological site} : {A selected site with geological
    findings, e.g., a {Volcanological site} or a
```

 ISBN: 978-1-61208-383-4

```
    {Meteorite site} , complementary to an
    {Archaeological site} .
    This site has been ...}

 + {Metorite site} : {A selected site with meteorite
    findings, e.g., meteorite crater, a special
    {Geological site} .
    This site has been ...}

 + {Volcanological site} : {A selected site with
    volcanological findings, e.g., volcanological
    findings like a volcano or fumarole, a special
    {Geological site} .
    This site has been ...}

 + Pottery    : {Archaeological site} major.countries {
  Italy France Spain Greece}

 + Stone      : {Geological site} major.countries {Italy
  France Spain Greece}
```

Figure 6. Excerpt of generated database matrix (lxworlddynamic).

Here, the matrix includes site and object types for the respective matrix with excerpts of descriptions and linked references.

B. Index selection and configuration

When a representation of matrix objects in dynamical spatial cartographic context is possible then selected objects can be integrated either from the matrix elements (e.g., sites) or from the context elements and references (e.g., cities, mounts, lakes, roads, rails, rivers, and grids). The listing (Figure 7) shows an excerpt of the generated index selection.

```
foreach i [geo::Names] {
   if {[lindex $geo::db($i) 0]=="city"} {
       "LX_World_database" $i : {city} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="mount"} {
       "LX_World_database" $i : {mount} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="site"} {
       "LX_World_database" $i : {site} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="lake"} {
       "LX_World_database" $i : {lake} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="road"} {
       "LX_World_database" $i : {road} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="rail"} {
       "LX_World_database" $i : {rail} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="river"} {
       "LX_World_database" $i : {river} loc [lrange $
        geo::db($i) 1 end]
   }
   if {[lindex $geo::db($i) 0]=="grid"} {
       "LX_World_database" $i : {grid} loc [lrange $
        geo::db($i) 1 end]
   }
}
```

Figure 7. Excerpt of generated index selection (lxworlddynamic).

Any part of the dynamically generated components can be individualised depending on context-sensitive attributes and workflow configuration. The listing (Figure 8) shows an example for the generated on-the-fly-symbol used for "sites".

```
set bitmaps(site) [image create bitmap -data [strimj::xbm
  "
....#....
...###...
..#...#..
.#.###.#.
##.###.##
.#.###.#.
..#...#..
...###...
....#...."] -foreground darkviolet]
```

Figure 8. Excerpt of generated on-the-fly symbol for sites (lxworlddynamic).

Different symbols can be integrated for different sites or for different groups. The objects with their symbols are only visible if the defined level, which is handled by the level handler, is active in the interactive view.

C. Dynamical visualisation and computation

The following image (Figure 9) shows a screenshot of a resulting dynamical visualisation of items in the result matrix, in this case the resulting archaeological context sites. The generated application utilises all the features so far described with the implementation.

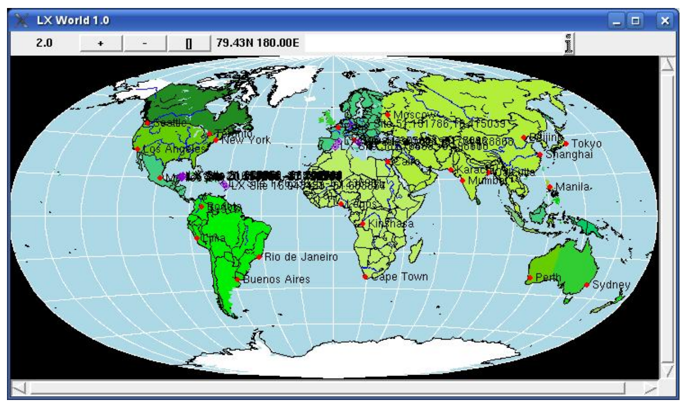

Figure 9. Archaeological context sites in interactive, dynamically generated spatial application (lxworlddynamic).

The screenshot illustrates the dynamical visualisation of the matrix elements for the context of the respective results. With the workflow a spatial context has been chosen for the matrix, creating the components. The spatial application component has been assembled by the workflow, integrating the object item references from the result matrix and secondary information from the referring knowledge resources objects with a dynamical and interactive view of the matrix.

Figure 10 shows a screenshot of a resulting dynamical zoom visualisation of matrix results and secondary information on geological and archaeological context sites. The partially shown superpositioning effect of the respective zoom is still visible in order to show the results, which can be separated in different cognitive views, different zooms, and event sensitive actions. The implementor can do anything with this feature he is interested in, e.g., use the interactive features for label stacks and level effects, being sensitive for single results or result group. Workflow sensitive cartographic material objects (e.g., cities, land, sea, countries, border lines, grid lines) are shown for orientation and context and support cognitive feature display. The shown zoom value and the scroll bars indicate the level of detail.

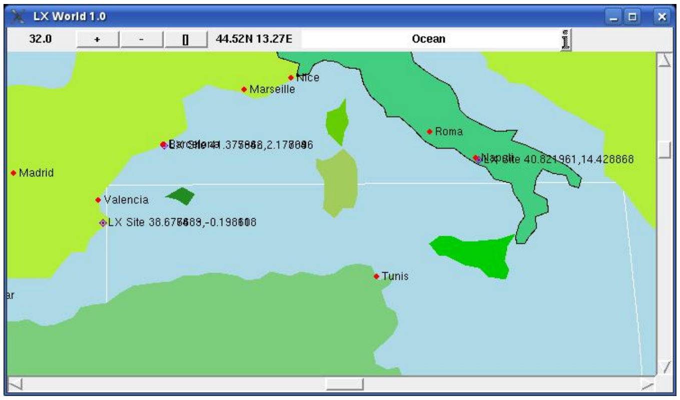

Figure 10. Zoom archaeological and geological sites and context, integrated in interactive, dynamically generated spatial application (lxworlddynamic).

The screenshot illustrates the matrix elements and their references within the knowledge resources. In this example, an active context-sensitive window component is delivering the secondary information. This component is actively communicating with the other components. The site entity data regarding levels and objects referred from the matrix elements is dynamically available (Figure 11).

```
LX Site 16.043153,-61.663374

level   2
:       site
loc     16.043153 61.663374
```

Figure 11. Example of a single site entity data extract (lxworlddynamic).

Here the matrix elements are referring to the attributes of the knowledge resources' objects, e.g., sites and cities. In this example, the displayed data excerpt includes the level, the type, and the location of the respective site. Figure 12 shows a screenshot of the corresponding dynamical site database.

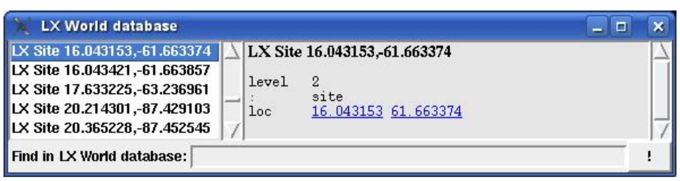

Figure 12. Site database, secondary information (lxworlddynamic).

The site database can be accessed by integrated or external applications' components, e.g., for searches, references or for generating further result matrices. The screenshot illustrates the matrix elements and their references within the knowledge resources. In this example, an active context-sensitive window is delivering the secondary information.

VI. Evaluation

Application components can be created and assembled dynamically from any workflow. Knowledge objects can be used efficiently with any dynamical components. Database interfaces can be used dynamically and efficiently with the components. Graphical User Interfaces (GUI) and Application Programming Interfaces (API) can be used most flexible on that dynamical base.

The information created from arbitrary numbers of resources' objects in this excerpt includes site labels, level information, category information, as well as the georeferences. The components can trigger any instances and events dynamically and interactively. This allows any kind of processing, computation, and visualisation, from sketch like visualisation to special cartographical mapping. All the more, the dynamical components based on the knowledge resources and the lxworlddynamic frameworks allow for the interactive, autonomous components' generation. The components have been successfully implemented on a number of operating systems (e.g., SuSE Linux, Debian, Red Hat, and Scientific Linux, as well with older and up-to-date distributions).

The knowledge and system architecture allows to seamlessly integrate with all the steps required for a sustainable implementation of the methodological bases. The case studies done over several decades of knowledge resources' creation and development and two decades of application component developments have shown that plans for extending structures, creating interfaces, self-learning components, container formats, and integration with life cycles and systems resources' operation can be assured even for long-term application.

VII. Conclusions

It has been demonstrated that with the proposed framework and concept context-sensitive dynamical components can be successfully created on base of universal knowledge resources. It has been shown that even standalone dynamical components can be created based on the implementation of the foundations, supporting arbitrary dynamical, modular, portable, and extendable database application programming and database graphical user interfaces. The implementation can utilise workflows and algorithms for knowledge discovery and selection up to intelligent application component creation. It integrates very efficiently in workflow chains, e.g., for computation and visualisation, and is very well usable even for rapid prototyping environments.

The integration is non-invasive regarding the knowledge resources for uni-directional visualisation and computation. If the intention with an application scenario is to update information consistently then multi-directional workflows can also update objects in side knowledge resources or containers from the created components in arbitrary ways, ensuring consistency and plausibility, as well as following management and security policies. Future work will be focussed on issues and usability beyond the plain Big Data approaches, resulting from data values, creating implementations supporting data vitality.

 ISBN: 978-1-61208-383-4

ACKNOWLEDGEMENTS

We are grateful to all national and international partners in the GEXI cooperations for the innovative constructive work. We thank the Science and High Performance Supercomputing Centre (SHPSC) for long-term support of collaborative research since 1997, including the GEXI developments and case studies. Special thanks go to the scientific colleagues at the Gottfried Wilhelm Leibniz Bibliothek (GWLB) Hannover, especially Dr. Friedrich Hülsmann, for prolific discussion within the long-term "Knowledge in Motion" (KiM) project, for inspiration, and practical case studies. Many thanks go to Mrs. Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, to the Institute for Legal Informatics (IRI), Leibniz Universität Hannover, and to the scientific colleagues at the Westfälische Wilhelms-Universität (WWU), for discussion, support, and sharing experiences on collaborative computing and knowledge resources and for participating in fruitful case studies, as well as to the participants of the postgraduate European Legal Informatics Study Programme (EULISP) for prolific discussion of scientific, legal, and technical aspects over the last years. Last but not least we thank Richard Suchenwirth for his initial implementation of Tclworld, which could be used as a tool for handling result matrices.

REFERENCES

[1] "LX-Project," 2014, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/#LX (Information) [accessed: 2014-10-05].

[2] E. Dodsworth and L. W. Laliberte, Eds., Discovering and using historical geographic resources on the Web: A practical guide for Librarians. Lanham: Rowman and Littlefield, 2014, ISBN: 0-8108-914-1.

[3] S. A. Keller and R. Schneider, Eds., Wissensorganisation und -repräsentation mit digitalen Technologien. Walter de Gruyter GmbH, 2014, Bibliotheks- und Informationspraxis, ISSN: 0179-0986, Band 55, ISBN: 3-11-031270-0.

[4] E. W. De Luca and I. Dahlberg, "Die Multilingual Lexical Linked Data Cloud: Eine mögliche Zugangsoptimierung?" Information Wissenschaft & Praxis, vol. 65, no. 4–5, 2014, pp. 279–287, Deutsche Gesellschaft für Information und Wissen e.V. (DGI), Ed., De Gruyter Saur, ISSN: 1434-4653, e-ISSN: 1619-4292, DOI: 10.1515/iwp-2014-0040, (title in English: The Multilingual Lexical Linked Data Cloud: A possible semantic-based access to the Web?).

[5] L. Schumann and W. G. Stock, "Ein umfassendes ganzheitliches Modell für Evaluation und Akzeptanzanalysen von Informationsdiensten: Das Information Service Evaluation (ISE) Modell," Information Wissenschaft & Praxis, vol. 65, no. 4–5, 2014, pp. 239–246, Deutsche Gesellschaft für Information und Wissen e.V. (DGI), Ed., De Gruyter Saur, ISSN: 1434-4653, e-ISSN: 1619-4292, DOI: 10.1515/iwp-2014-0043, (title in English: A comprehensive holistic model for evaluation and acceptance analyses of information services: The Information Service Evaluation (ISE) model).

[6] G. Isaew and A. Roganow, "Qualitätssteuerung von Informationssystemen: Theoretisch-methodologische Grundlagen," Information Wissenschaft & Praxis, vol. 65, no. 4–5, 2014, pp. 271–278, Deutsche Gesellschaft für Information und Wissen e.V. (DGI), Ed., De Gruyter Saur, ISSN: 1434-4653, e-ISSN: 1619-4292 DOI: 10.1515/iwp-2014-0044, (title in English: Quality Management of Information Systems: Theoretical and methodological basics).

[7] "Geo Exploration and Information (GEXI)," 1996, 1999, 2010, 2014, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI [accessed: 2014-10-05].

[8] C.-P. Rückemann, "Active Map Software," 2001, 2005, 2012, URL: http://wwwmath.uni-muenster.de/cs/u/ruckema (information, data, abstract) [accessed: 2012-01-01].

[9] "Europeana," 2014, URL: http://www.europeana.eu/ [accessed: 2014-11-30].

[10] "WDL, World Digital Library," 2014, URL: http://www.wdl.org [accessed: 2014-11-30].

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

[12] C.-P. Rückemann, "Knowledge Integration for Scientific Classification and Computation," in The Fourth Symposium on Advanced Computation and Information in Natural and Applied Sciences, Proceedings of The 12th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 22–28, 2014, Rhodes, Greece, Proceedings of the American Institute of Physics (AIP), AIP Conference Proceedings. AIP Press, 2014, ISSN: 0094-243X.

[13] C.-P. Rückemann, "Long-term Sustainable Knowledge Classification with Scientific Computing: The Multi-disciplinary View on Natural Sciences and Humanities," International Journal on Advances in Software, vol. 7, no. 1&2, 2014, pp. 302–317, ISSN: 1942-2628.

[14] "GMT - Generic Mapping Tools," 2014, URL: http://imina.soest.hawaii.edu/gmt [accessed: 2014-01-12].

[15] 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, pp. 235–255, ISSN: 1942-2679, LCCN: 2008212456 (Library of Congress), URL: http://www.iariajournals.org/intelligent_systems/intsys_v6_n34_2013_paged.pdf [accessed: 2014-11-30].

[16] C.-P. Rückemann, "Enabling Dynamical Use of Integrated Systems and Scientific Supercomputing Resources for Archaeological Information Systems," in Proc. INFOCOMP 2012, Oct. 21–26, 2012, Venice, Italy, 2012, pp. 36–41, ISBN: 978-1-61208-226-4.

[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: 2014-11-30].

[18] "UDC Online," 2014, http://www.udc-hub.com/ [accessed: 2014-10-05].

[19] 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.

[20] "Tcl Developer Site," 2014, URL: http://dev.scriptics.com/ [accessed: 2014-11-30].

[21] "Open-MPI," 2014, URL: http://www.open-mpi.org [acc.: 2014-06-29].

[22] R. Suchenwirth, "Tclworld, Version 0.6, Konstanz, 2002-01-31," 2002, URL: http://wiki.tcl.tk/_repo/wiki_images/tm06.zip [acc.: 2014-11-30].

[23] R. Suchenwirth, "dbapi – A simple database API," 2002, URL: http://wiki.tcl.tk/_repo/wiki_images/tm06.zip [accessed: 2014-11-30].

[24] R. Suchenwirth, "dbgui.tcl – A little database GUI," 2002, URL: http://wiki.tcl.tk/_repo/wiki_images/tm06.zip [accessed: 2014-11-30].