Institute of Architecture of Application Systems

The SustainLife Project – Living Systems in Digital Humanities

Claes Neuefeind[1], Lukas Harzenetter[2], Philip Schildkamp[1], Uwe Breitenbücher[2],Brigitte Mathiak[1], Johanna Barzen[2], and Frank Leymann[2]

[1]Data Center for the Humanities (DCH), University of Cologne, Germany
{claes.neuefeind,philip.shildkamp,brgitte.mathiak}@uni-koeln.de

[2]Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{barzen,breitenbuecher,harzenetter,leymann}@iaas.uni-stuttgart.de

BibTeX

```
@inproceedings{Neuefeind2018-SummerSoC,
  Author    = {Claes Neuefeind and Lukas Harzenetter and Philip Schildkamp and
               Uwe Breitenb{\"u}cher and Brigitte Mathiak and Johanna Barzen
               and Frank Leymann},
  Title     = {{The SustainLife Project – Living Systems in Digital Humanities}},
  Booktitle = {Papers From the 12th Advanced Summer School of Service-Oriented
               Computing (SummerSOC 2018)},
  Publisher = {IBM Research Division},
  Pages     = {101--112},
  Month     = oct,
  Year      = 2018,
}
```

The full version of this publication has been presented as a poster at the Advanced Summer School on Service Oriented Computing (SummerSOC 2018).
http://www.summersoc.eu

University of Stuttgart
Germany

The SustainLife Project – Living Systems in Digital Humanities

Claes Neuefeind[1], Lukas Harzenetter[2], Philip Schildkamp[1], Uwe Breitenbücher[2], Brigitte Mathiak[1], Johanna Barzen[2], and Frank Leymann[2]

[1] University of Cologne, 50923 Cologne, Germany
[2] University of Stuttgart, 70569 Stuttgart, Germany
{c.neuefeind,philip.schildkamp,bmathiak}@uni-koeln.de
{lukas.harzenetter,uwe.breitenbuecher,
johanna.barzen,frank.leymann}@iaas.uni-stuttgart.de

Abstract. In the arts and humanities, research applications play a central role in securing and presenting digital results. However, due to their steadily increasing number and their heterogeneity, it is difficult to ensure the sustainability and durability of this kind of living systems from an organizational point of view. This paper describes a project for the preservation of specialized web-based research applications in the humanities. The SustainLife project investigates to what extent methods and technologies of professional cloud deployment and provisioning strategies can be applied to problems of long-term availability of research software as they are omnipresent in humanities data centers such as the Data Center for the Humanities (DCH) at the University of Cologne. Technological basis of the project is the OASIS standard TOSCA and the Open Source implementation OpenTOSCA, respectively, which was developed at the Institute for Architecture of Application Systems (IAAS) at the University of Stuttgart. In the course of the project selected use cases from the field of Digital Humanities (DH) will be modeled in TOSCA to be able to automatically deploy them upon request at any time. The TOSCA standard enables a portable description of the modeled systems independent of specific providers to facilitate their long-term availability. The aim is to provide system components described in the use cases in a component library, as well as in the form of TOSCA-compliant application templates to make them available for reuse in other DH projects.

Keywords: Living Systems, Sustainability, Research Software.

1 Introduction

The exponential growth and the increasing use of digital research data have a significant impact on the research process in the humanities. To take advantage of the benefits of digitization, appropriate infrastructure must be created that guarantees the management of research data, its permanent availability, and free access. The large number of scientific recommendations, stocktakings, surveys, and institutional guidelines on the subject of research data, which have been published in recent years, are signs of increasing awareness of the problem, political willingness to act, but also of a continuing need for action [18, 23, 7, 19]. Existing European and national infrastructure projects in the humanities (e.g., CLARIN and DARIAH) as well as the establishment of subject-specific data centers such as, for example, the Data Center for the Humanities (DCH) of the University of Cologne (see http://dch.uni-koeln.de), have steadily improved the supply situation for research data. However, not all of the produced research results are actually made available for reuse or, respectively, are equipped for permanent availability in a highly dynamic digital world [20, p. 31, 23, p. 54].

Up to now, the discussion has mostly focused on information systems for the standardized storage and provision of primary research data. It is largely overlooked that the majority of the digital products in the humanities do not only consist of primary research data, but are, above all, available in the form of research software. In fact, digital systems such as research databases, digital editions, digital presentation systems, interactive visualizations, and virtual research environments – to name only a few – represent an essential component of the research results, especially in the context of Digital Humanities (DH). Often, these are the actual bearers of information content or, respectively, of the added value of the scientific output provided in the project [22]. A sustainability policy that falls back onto the separation and archiving of the primary data alone inevitably leads to the loss of information, and in the worst case reduces the scientific benefit to zero [1]. In contrast to traditional forms of securing results, published books for instance, the permanent maintenance, support and provisioning of such “living systems” [21] continues to be a major technical, organizational, and thereby ultimately financial challenge. While it is comparatively easy to preserve data sets of primary research data in data archives for posterity, living systems are part of a digital ecosystem and must regularly adapt to it, e.g., in the form of updates.

1.1 Problem Statement

Humanities data centers face the challenge of preserving an unknown, potentially unlimited number of research software, in order to assure their availability on a permanent basis. Internal evaluations from the consulting activities of the DCH at the Faculty of Arts and Humanities of the University of Cologne show that more than one third of all inquiries aim at the creation of individual research platforms [1]. In a survey on research data among the scientific staff at the Faculty of Arts and Humanities conducted in 2016, 62% of the respondents said that they need support for their running applications [11]. It is therefore all the more surprising that there have been only few dedicated studies so far on the exact nature of these living systems. Wuttke et al. [22] found that “research data types” in humanities can be categorized into three types of applications, namely (i) digital editions, (ii) databases, and (iii) interactive visualizations [22]. A systematic survey of all humanities research applications is still missing, both in the general research landscape and at the Faculty of Arts and Humanities. Not only a high methodological diversity is to be expected, but also a considerable diversity of the technologies used.

This heterogeneity is a major challenge for a sustainable conservation strategy. Research applications are by no means static objects, but are subject to continuous change. For example, many applications are platforms that accept user input and, thus, constantly change their database. Browser updates and changes in usage habits can make certain components unusable or obsolete and, thus, trigger the need for a serious code revision. Vital security updates regularly require actions to be taken and can result in cascades of further updates and software adaptations. To ignore continuous maintenance may save costs in the short term, but seriously increases the problem in the medium and long term. Experiences from the "LAZARUS Project" carried out at the DCH in 2013-2015 show that applications that remain without maintenance for a longer period of time can only be revitalized at a great expense of resources [6].

Furthermore, data centers are confronted with the dilemma that limited project durations make permanent operation significantly more difficult from an organizational point of view. This is particularly fatal in humanities, where often a different measure of sustainability is applied than in the comparatively fast-moving natural sciences. The institutional provision of research data for a few decades fulfills the purpose of verifiability and reproducibility in the sense of good scientific practice; but for the provision of cultural heritage objects, a time limit does not even make sense. Even if a shutdown cannot be avoided for safety or cost reasons, it must at least be ensured that the application and the data basis are archived in a way that they can be restored loss-free and reusable at any time.

1.2 Key Challenges

In summary, we see the following challenges in connection with living systems, that have to be addressed to establish a strategy for their sustainable preservation:

1. So far there is no overview of the overall landscape of research applications in the Digital Humanities, which is a necessary prerequisite for dealing with the problem.
2. A quick analysis of use cases from the field of DH shows a wide range of technologies and methodological approaches. In contrast to large commercial information systems, the field of DH is characterized by a great number of smaller, highly heterogeneous software solutions, which are all equally subject to the problem of "software aging" [17].
3. The exposure of these technologies to the Internet is inherently associated with continuous development efforts, e.g., to keep the systems permanently available through security updates.
4. However, funding for long-term software maintenance is usually very limited.
5. If a service is not needed anymore, its current application state must be stored so that it can be reinstated in the same state upon request during the next provisioning. Both concepts are currently missing.
6. Starting (research) applications ten years after their last successful provisioning, e.g. to verify the research results, does not work in most cases. Especially, if the underlying infrastructure has changed to current state-of-the-art components because of security and availability issues.

2 The SustainLife Project

Against the background of the challenges described above, the Institute for Architecture of Application Systems (IAAS) of the University of Stuttgart and the Data Center for the Humanities (DCH) of the Faculty of Arts and Humanities, University of Cologne, started the joint project “SustainLife” in March 2018, funded by the DFG under the funding line “Scientific Library Services and Information Systems” (LIS) for a period of three years. The project develops solutions based on the OASIS standard TOSCA [14, 16]. TOSCA provides a portable description of IT systems in the form of models to automate their provisioning and management which is described in more detail in Section 2.1. This enables the creation of adaptable and future-proof software architectures. In the following sections 2.1 and 2.2, the technical basis (TOSCA and OpenTOSCA) of the project is explained in more detail. The main objectives of the project are then described in Section 2.3, followed by the projects’ methodology in Section 2.4.

2.1 TOSCA

The Topology Orchestration Specification for Cloud Applications (TOSCA) [14, 15, 16] is an OASIS standard for modelling, provisioning, and managing cloud applications in a standardized and provider-independent way. In TOSCA, a cloud application or service is modelled in a “Service Template”. Inside a Service Template, the “Topology Template” describes the service’s topology as a directed multigraph, whereby the nodes are represented by “Node Templates” and “Relationship Templates” detail the edges. Underneath, TOSCA defines a type system defining common properties and attributes in “Node Types” and “Relationship Types” respectively. For example, a web application written in PHP may be hosted on an Apache webserver which itself is hosted on an Ubuntu virtual machine. Therefore, three Node Types, “web-application”, “Apache-server”, and “Ubuntu-VM”, as well as the “hosted-on” Relationship Type must be available. A Service Template describing this application will contain three Node Templates, “app”, “webserver”, and “VM”, and two Relationship Templates modelling the relationship between the app and the webserver, as well as the webserver and the virtual machine – where each template is an instance of the respective type definition.

In general, TOSCA provides various extensions and mechanisms for modeling, provisioning and managing any type of software component. In addition, existing technologies, such as container systems like Docker, can be seamlessly integrated. TOSCA is therefore not a competitive approach to existing technologies, but a way to combine them [12]. However, in contrast to Docker and Kubernetes, TOSCA enables the deployment of arbitrary software components such as legacy software [12].

To automatically deploy, provision and manage the modelled service, TOSCA defines a self-contained archive called “Cloud Service Archive” (CSAR) which contains the Service Template, all Node Types and Relationship Types, as well as all required software artifacts, scripts, and binaries required for provisioning. A TOSCA runtime environment can consume a CSAR to automatically deploy and instantiate the enclosed application. For provisioning a service, TOSCA supports both, declarative and imperative [8] deployments using plans like BPMN4TOSCA [9].

2.2 OpenTOSCA

In a series of research projects, the University of Stuttgart has developed the OpenTOSCA ecosystem – an open source implementation for the TOSCA standard, which will be adapted to the specific needs of the DH and which will form the basis for the modelling of use cases within the SustainLife project. The OpenTOSCA ecosystem includes (i) the modeling tool Winery, which enables the creation of TOSCA-based application models [10], (ii) the runtime environment OpenTOSCA for automated provisioning and management of the modelled applications [4], and (iii) the self-service portal Vinothek [5], which lists all applications installed in the OpenTOSCA container and serves as a graphical interface to the user.

In a first publication by Breitenbücher et al. [3] it was already shown that the TOSCA standard is generally suitable for assuring the digital sustainability of research results [3], as research applications, which are packaged in CSARs, can be executed years later by a TOSCA runtime environment. However, there is currently no possibility to update the service's components. If a component must be exchanged because of security issues, or because it is not available anymore, the CSAR may no longer be deployable. This currently limits the use of CSARs for living systems.

2.3 Objectives

The overall goal of the project is to improve the sustainability of living systems in the Digital Humanities. Against the background of the technological requirements associated with the TOSCA standard, the subordinated goals of the project result directly from the challenges described above:

1. Our first goal is to provide a systematic overview of the field of DH that goes beyond previous literature, particularly in the area of technological and methodological classification employed in the software artifacts.
2. Investigating the multitude of possible solutions, a set of key components that are frequently used and that consequently have high potential for synergetic effects will be identified and modeled in TOSCA artifacts.
3. The project extends the OpenTOSCA ecosystem to include application templates, component types, and to implement additional management functionalities which provide standardized operating and maintenance solutions for these components, e.g., applying (security) updates or software patches.
4. Additionally, concepts are needed which are able to "freeze" and "defrost" an application to ensure that they can be reinstated in the same application state as they were decommissioned.
5. We expect this to reduce maintenance costs and will evaluate this expectation on the basis of selected use cases. Gained experiences and best practices will be fed back to the community, e.g., in workshops and publications.

2.4 The Sustainability

To achieve these goals, it is essential to first obtain an overview of the specific needs of living systems in the Digital Humanities in detail; for example, which technologies are actually being used and how can they be automated to a greater extent using the TOSCA standard. This is meant to give a general idea of the technological needs and the potential of the methodology in the context of real use cases. Based on this requirement analysis, a set of concrete use cases is selected and a concrete implementation plan is created for each of them.

On the basis of the concrete use cases, frequently used key components with great synergetic potential, typical application structures and central maintenance tasks are identified. First, the components fundamentally required to implement the selected use cases – such as web servers or operating systems – are modeled using TOSCA in order to use them for automated provisioning and sustainable maintenance. For example, the Spring framework, and various databases such as MongoDB, have to be deployable by the OpenTOSCA ecosystem. The components can then be used in further use cases and are intended to simplify future application developments, as they can be reused when modelling other applications in TOSCA.

For automated provisioning and maintenance of applications using TOSCA, associated models describing the structure of the application must be developed. This is done in close cooperation with the use case partners. In the process of modelling the use case applications, the expenses incurred and savings made are recorded, e.g. which problems occur and which requirements still exist, so that these can be processed further in the further course of the project.

To facilitate an easier application development using TOSCA and, respectively, the description of existing applications in TOSCA, model templates are developed which can be reused as a basis. For this purpose, the use cases are analyzed for typical application structures and corresponding models are created. For example, a common pattern for web applications is to use LAMP-based technologies that use **L**inux-based operating systems to run **A**pache web servers and **M**ySQL databases to run **P**HP/**P**erlin/**P**ython-based applications. Common structures of this kind should be covered by the templates. The aim of these model templates is therefore to develop new applications efficiently and with little effort using TOSCA and the OpenTOSCA ecosystem and to maintain them with the help of the management concepts developed in this project. In addition, a number of further extensions of the OpenTOSCA ecosystem are necessary for their modelling, which will be carried out in the further course of the project. These are explained in section 4.

3 Use Cases

Up to this stage of the project, four potential use cases were selected for the project as examples of “living systems”, all originating from the Faculty of Arts and Humanities of the University of Cologne. The selected use cases illustrate the diversity of the technologies used and the high degree of methodological and content-related specialization that is typical for the DH research community.

3.1 Digital Romansh Chrestomathy

Our first use case is the "Digital Romansh Chrestomathy" (DRC, see http://www.crestomazia.ch), which contains typical DH technologies like XML as well as very specific tailor-made system components. It is therefore a representative for most DH software components, as it combines several different problems existing in common DH software. The DRC was a DFG-funded cooperation project of the Department for Linguistic Information Processing and the University and City Library of Cologne between 2009 and 2011. The aim of the DRC project was to create a text corpus based on Caspar Decurtins' "Rätoromanische Chrestomathie" (RC). With its approximately 7,500 pages of text from four centuries and the coverage of the five main idioms, the RC is considered the most important collection of texts for the Romansh language. The project implemented a virtual research environment consisting of an Eclipse-based editor and a XML database, as well as a portal page for searching the texts [12]. There are still active users of the editor who contribute corrections and comments, and the search functionalities are also still in use. Since project completion in 2011, the application had to be adapted several times to ensure its usability.

Technologies (selection): Eclipse-RAP, eXist-db, Java, SCALA

3.2 Digital Averroes Research Environment

The "Digital Averroes Research Environment" (DARE, http://dare.uni-koeln.de) is a digital research platform of the Thomas Institute of the University of Cologne. The aim of the portal is to make the complete work of the philosopher Averroës publicly accessible. In recent years, a large number of manuscripts and print editions have been digitized, scientifically encoded on the basis of the TEI XML standard, and were published via the DARE portal. The current DARE platform uses the Oxygen XML-editor for data input and provides a custom search implemented in Free Pascal and XSL-generated visualizations. In the future, the DARE web application will not only provide a search and research functionality, but also the possibility to annotate the published material by external users. The application is currently being refactored to provide a browser-based user interface based on state-of-the art web technologies.

Technologies (selection): XSL, Free Pascal, Oxygen, Express-js, Nodejs, MySQL

3.3 Vedaweb

Subject of the cooperation project "VedaWeb" (see http://vedaweb.uni-koeln.de), involving the department of Linguistics, the Cologne Center for eHumanities (CCeH) and the Data Center for the Humanities (DCH) is a web-based platform for the linguistic research of old Sanskrit texts. VedaWeb will make it possible to view the texts in digital format as well as to search through them based on lexical and corpus linguistic criteria. The project builds upon the Rigveda, one of the oldest and most important texts of the Indo European language family which was composed in Vedic i.e., the oldest form of Sanskrit in late second millennium B.C. In the course of the project, various research and analytical tools will be developed and integrated into the

VedaWeb platform; for example, a combined search function according to linguistic parameters (lemma, word forms, morphological and metrical information), access to various translations and commentaries, as well as the possibility to export retrieved texts in TEI format according to user defined criteria. Of central importance is the linking of the text to the Digital Sanskrit Dictionaries Cologne (see http://www.sanskrit-lexicon.uni-koeln.de), which are hosted by the CCeH.

Technologies (selection): Spring.io, elasticsearch, mongoDB, Javascript, TEI

3.4 German Early Cinema Database

Our fourth use case is the "German Early Cinema Database" (see http://earlycinema.uni-koeln.de), which is maintained by the Institute of Media Culture and Theatre at the University of Cologne. The German Early Cinema Database contains data related to film supply, distribution, exhibition and reception in Germany between 1895 and 1926. The database consists of four parts: (i) approx. 5,000 texts on early cinema in Germany between 1895 and 1914, from a wide selection of non-film sources, like specialist journals and general newspapers. (ii) Information on itinerant and fairground cinemas in Germany and neighboring countries between 1896 and 1926. (iii) Information on approx. 45,000 films available on the German market between 1895 and 1920 (irrespective of their country of origin). (iv) A sample of film programs from 1905 to 1914, mostly from permanent cinemas compiled from the newspapers of nine German cities of different regions and sizes (approx. 1,200 programs from approx. 100 cinemas, containing approx. 3,800 different films). The implementation is based on a combination of CakePHP and MySQL. Due to the age of the application, the technologies used are no longer supported by the local data processing center in cologne. Since the German Early Cinea Database is an absolutely unique resource for this type of information, it is a typical example to illustrate the need to preserve this kind of applications.

Technologies (selection): CakePHP, MySQL, JavaScript

4 Work Packages

Existing work in the TOSCA environment already contributes to the goal of modelling and preserving applications for an instantiation after several years. However, most of the work did not consider living systems and their resulting requirements. During the SustainLife project, we want to address these requirements and extend the OpenTOSCA ecosystem as depicted in **Fig. 1**. The already existing components Winery and OpenTOSCA Container will be extended by the green elements in the respective work packages (WP) during the project. While boxes describe extensions to the software components, cylinders represent repositories containing, for example, the artifacts which are describing the aforementioned use cases in TOSCA. The work packages wrapping the planned extensions are described in the following sections.

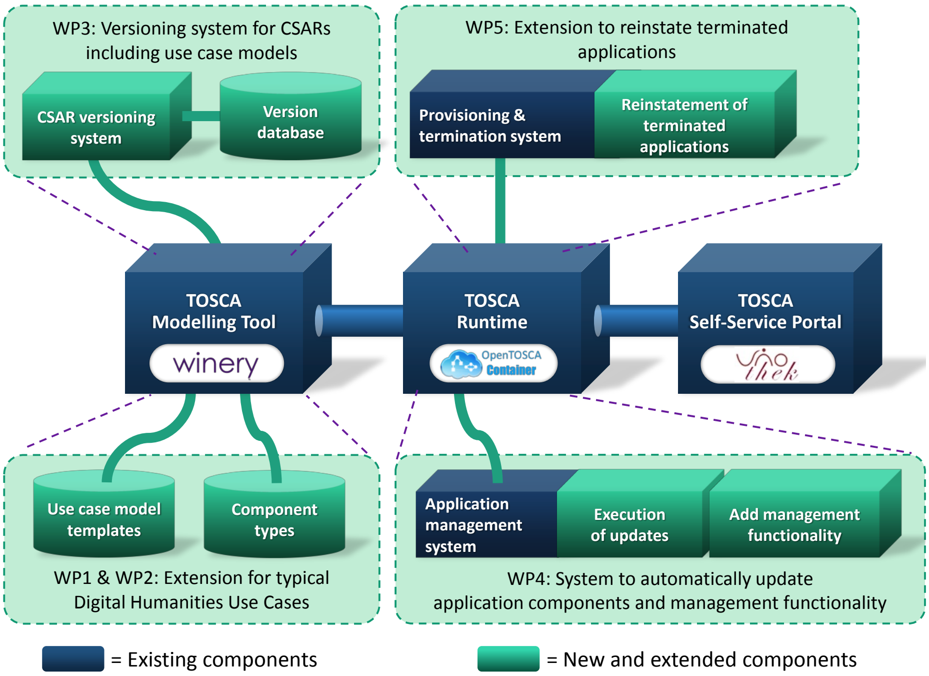

Fig. 1. Work plan of the SustainLife project

4.1 Development of Templates for Typical Use Cases (WP1 & WP2)

TOSCA depends on a generic type system enabling the reuse of recurring components like webservers, operating systems, or messaging middleware. Within the first two work packages, we will identify the components with the highest synergetic effects and model them as TOSCA artifacts to enable their automated provisioning using the OpenTOSCA Container. Examples for components which were already identified in the presented use cases are an Ubuntu virtual machine, Java runtime environments, and several types of databases like MySQL and mongoDB.

4.2 Versioning of TOSCA Models (WP3)

Living Systems are subject to constant changes. Webservers, for example, must be updated or even exchanged regularly. Furthermore, if an application is deployed ten years after its last successful provisioning, many components will be outdated and should not be used because of, e.g., security or legal issues. Most of these changes are required because new security issues become public and need to be fixed to ensure that no attacker can gain access to the server or the data. Therefore, a versioning concept for TOSCA models is required to maintain an overview over all available versions of the components modeled in TOSCA. Since there is no versioning mechanism for TOSCA available yet, we will work on an approach in the context of the Sustain-Life project.

Besides the management of security updates and patches, versioning of TOSCA artifacts is also required in a more general sense. For instance, it is an important prerequisite to track the evolution of a component over time. By comparing different versions of a service or component, it is possible to reproduce the performed steps and modifications. Beyond that, versioning also yields the possibility to maintain multiple service architectures of a single application. For example, if a service was designed to run on a local infrastructure in one version, another version can describe the topology in a public cloud setting.

4.3 Modification of Management Functionality (WP4)

Currently, TOSCA supports two kinds of implementing management functionality: (i) management operations at Node Types, and (ii) separately defined management plans. In TOSCA, every component type can define custom management functionality by defining management interfaces providing different operations. For example, TOSCA describes a lifecycle interface with the operations install, configure, start, stop, and uninstall [15, 16]. The operations can be implemented, for instance, by shell scripts performing the desired operation. More complex management functionality can be described and executed by management plans. The process of provisioning a whole service can be modeled with a management plan describing every step of the process which can be executed automatically by a TOSCA runtime. OpenTOSCA's Winery supports both flavors and is able to generate management plans for the provision and termination of an application [2].

There is one major constraint, however: all management functionality must be known at modelling time. While the installation of security updates is a known functionality which can be considered during modelling time, new and unforeseen management operations cannot be added during runtime which is a requirement coming from the living systems to remain sustainable. Therefore, we want to set our research focus in work package four on developing concepts to extend the management functionality of software components during their runtime.

4.4 Concepts to Reinstate Terminated Applications (WP5)

In the context of living systems, there is the requirement for stopping and reinstating applications at any subsequent point in time. In this context, the Digital Romansh Chrestomathy (DRC) presented above serves as an example. The DRC service provides a web-based editor building upon the Eclipse-RAP framework enabling users to correct the digital version of the Romansh Chrestomathy. After the remediation is completed, the editor should still be available for documentation purposes. Since the editor may not be used for some time, it should be possible to stop the application and reinstate it in the same application state to save computing resources and consequently costs. To achieve this, the application state of the DRC service must be stored and recovered during reinstatement.

Currently, neither the OpenTOSCA ecosystem, nor other cloud management systems supporting the TOSCA standard provide a functionality for saving and restoring the application state of services which are modeled and instantiated using TOSCA in a generic way. Consequently, the fifth work package in SustainLife is dedicated to the development of concepts and functionalities to support freezing and defrosting of stateful services using TOSCA and especially the OpenTOSCA ecosystem.

5 Summary

During the SustainLife project we want to focus our research towards supporting living systems in the Digital Humanities using TOSCA. The overall objective of the project described here is to develop generic concepts for standards-based operation and maintenance solutions and to implement them for specific components and application structures in a way that they can find practical application in humanities data centers like the DCH. One major shortcoming we want to address is that CSARs should still be deployable ten years after their development. Therefore, approaches to freeze and defrost whole applications, as well as updating used components to state-of-the-art ones, will be in the focus of our research in SustainLife. Findings and best practices from the project are prepared in a way that solution models can be transferred to partners and other data centers, and are communicated to the scientific public community through workshops and publications.

Acknowledgements

This work is partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

References

1. J. Blumtritt, B. Mathiak, “Consulting Workflow for Humanities Research Data”, In: Forschungsdaten in den Geisteswissenschaften (FORGE 2016), Hamburg, 2016.
2. U. Breitenbücher, T. Binz, K. Képes, O. Kopp, F. Leymann, J. Wettinger. “Combining Declarative and Imperative Cloud Application Provisioning based on TOSCA”, IC2E, S. 87–96, 2014.
3. U. Breitenbücher, J. Barzen, M. Falkenthal, F. Leymann. “Digitale Nachhaltigkeit in den Geisteswissenschaften durch TOSCA: Nutzung eines standardbasierten Open-Source Ökosystems”. Konferenzabstracts DHd 2017: Digitale Nachhaltigkeit, S. 235-237, 2017.
4. T. Binz, U. Breitenbücher, F. Haupt, O. Kopp, F. Leymann, A. Nowak, S. Wagner. “OpenTOSCA - A Runtime for TOSCA-based Cloud Applications”. In: ICSOC, 2013, S. 692-695, 2013.
5. U. Breitenbücher, T. Binz, O. Kopp und F. Leymann. “Vinothek - A Self-Service Portal for TOSCA”. In: ZEUS 2014, p. 69-72, 2014.
6. S. Bingert, J. Blumtritt, S. Buddenbohm, C. Engelhardt, S. Kronenwett, D. Kurzawe. “Anwendungskonservierung und die Nachhaltigkeit von Forschungsanwendungen” In: Forschungsdaten in den Geisteswissenschaften (FORGE 2016), Hamburg, 2016.

7. DV-ISA. “Umgang mit digitalen Daten in der Wissenschaft: Forschungsdatenmanagement in NRW. Eine erste Bestandsaufnahme”, 2016, Version 0.7, https://www.dvisa-nrw.de/veroeffentlichungen/veroeffentlichungen-container-oeffentlich/dv-isa-vorstudie-bestandsaufnahme-forschungsdatenmanagement
8. C. Endres, Breitenbücher, U., Falkenthal, M., Kopp, O., Leymann, F., & Wettinger, J. “Declarative vs. Imperative: Two Modeling Patterns for the Automated Deployment of Applications”. In Proceedings of the 9th International Conference on Pervasive Patterns and Applications (PATTERNS), 2017, pp. 22-27
9. O. Kopp, T. Binz, U. Breitenbücher, F. Leymann. „BPMN4TOSCA: A domain-specific language to model management plans for composite applications”. In International Workshop on Business Process Modeling Notation, pp. 38-52. Springer, Berlin, Heidelberg, 2012.
10. O. Kopp, T. Binz, U. Breitenbücher und F. Leymann. “Winery – A Modeling Tool for TOSCA-based Cloud Applications”. In: ICSOC, 2013, S. 700-704, 2013.
11. S. Kronenwett. “Forschungsdaten an der Philosophischen Fakultät der Universität zu Köln” (Kölner Arbeitspapiere zur Bibliotheks- und Informationswissenschaft, Bd. 78), 2017.
12. F. Leymann, U. Breitenbücher, S. Wagner, J. Wettinger. “Native Cloud Applications: Why Monolithic Virtualization Is Not Their Foundation”. Cloud Computing and Services Science, Springer, pp. 16-40, 2017.
13. C. Neuefeind, J. Rolshoven, F. Steeg. “Werkzeuge und Verfahren für die Korpuserstellung durch kollaborative Volltexterschließung”. In: Conference of the German Society for Computational Linguistics and Language Technology (GSCL), Hamburg, pp. 163-168, 2011.
14. OASIS: “Topology and Orchestration Specification for Cloud Applications Version 1.0”, 2013.
15. OASIS. “Topology and Orchestration Specification for Cloud Applications (TOSCA) Primer Version 1.0”. 2013.
16. OASIS: “TOSCA Simple Profile in YAML”, Version 1.0, 2015.
17. D. L. Parnas. “Software Aging”. In: Proceedings of the 16th International Conference on Software Engineering (ICSE 1994). IEEE, May 1994, pp. 279-287, 1994.
18. H. Pampel, R. Bertelmann. “Data Policies im Spannungsfeld zwischen Empfehlung und Verpflichtung”. In: Handbuch Forschungsdatenmanagement, 2011, pp. 49–61.
19. Rat für Informationsinfrastrukturen. “Leistung aus Vielfalt. Empfehlungen zu Strukturen, Prozessen und Finanzierung des Forschungsdatenmanagements in Deutschland”, 2016, http://www.rfii.de/download/rfii-empfehlungen-2016/
20. M. Razum, J. Neumann. “Das RADAR Projekt: Datenarchivierung und -publikation als Dienstleistung - disziplinübergreifend, nachhaltig, kostendeckend”. In: o|bib Das offene Bibliotheksjournal, 1/1, 2014, pp. 30–44, https://www.o-bib.de/article/view/2014H1S30-44/117
21. P. Sahle, S. Kronenwett. “Jenseits der Daten: Überlegungen zu Datenzentren für die Geisteswissenschaften am Beispiel des Kölner Data Center for the Humanities”. In: LIBREAS. Library Ideas 23, pp. 76-96, 2013.
22. U. Wuttke, C. Engelhardt, S. Buddenbohm. “Angebotsgenese für ein geisteswissenschaftliches Forschungsdatenzentrum”. In: Zeitschrift für digitale Geisteswissenschaften, 2016.
23. Wissenschaftsrat. “Empfehlungen zur Weiterentwicklung der wissenschaftlichen Informationsinfrastrukturen in Deutschland bis 2020”, 2012, http://www.wissenschaftsrat.de/download/archiv/2359-12.pdf