Digital Preservation Of Business Processes with TIMBUS Architecture

Mykola Galushka
SAP (UK) Ltd
The Concourse, Queen's Road,
Queen's Island, Titanic Quarter,
Belfast, BT3 9DT
mykola.galushka@sap.com

Philip Taylor
SAP (UK) Ltd
The Concourse, Queen's Road,
Queen's Island, Titanic Quarter,
Belfast, BT3 9DT
philip.taylor@sap.com

Wasif Gilani
SAP (UK) Ltd
The Concourse, Queen's Road,
Queen's Island, Titanic Quarter,
Belfast, BT3 9DT
wasif.gilani@sap.com

John Thomson
CMS
Portugal
john.thomson@caixamagica.pt

Stephan Strodl
Secure Business Austria
Vienna, Austria
sstrodl@sba-research.org

Martin Alexander Neumann
KIT, Kaiserstrasse 12, 76131
Karlsruhe,Germany
mneumann@teco.edu

ABSTRACT

The majority of existing digital preservation solutions are focusing on the long-term storage of digital content such as documents, images, video, audio files and other domain specific data. Preservation of an Information Technology infrastructure for supporting business processes is a much more challenging task. It requires the preservation of software and hardware stacks as well as relevant contexts, which together, provide an execution layer for running business processes. The proposed TIMBUS architecture addresses limitations of existing digital preservation solutions and provides a complete framework for preserving business processes implemented upon a service oriented architecture.

Keywords

digital preservation, business process

1. INTRODUCTION

Providing business continuity is an important task facing many multi-national companies. Failure to provide business continuity often leads to substantial financial losses and may cause total or partial loss of business. The most common approach to address potential adverse circumstances or events is to provide effective Business Continuity Management (BCM). The main objective of BCM is to access different risk factors and establish contingency plans for minimising impact of potential hazardous events on existing business processes (BPs).

Many BPs are heavily dependent on sophisticated information technology (IT) infrastructure, which is supporting different business transactions and providing critical information for making important business decisions. Constant evolution of the technology landscape brings new challenges for any business organisation to support their IT infrastructure, where new hardware and software solutions are developed and adopted on a regular basis. These new versions of hardware and software components often have backward compatibility limitations. In some contexts digital preservation (DP) is one of the most effective solution for dealing with evolving digital infrastructures.

The majority of existing DP solutions focus on the long-term storage of digital content such as documents, images, video and audio files. Preservation of an IT infrastructure for supporting BPs is a much more challenging task. It requires the preservation of software and hardware stacks as well as relevant contexts, which together, provide an execution layer for running BPs.

DP architecture developed within the TIMBUS[1][12] project provides a unique set of solutions going beyond the scope of existing DP approaches. It covers all aspects of traditional DP system such as preserving a digital content but also addresses enterprise risk analysis and business continuity planning. It covers a wider scope of DP processes, which includes Intelligent Enterprise Risk Management (IERM) for automatic identification and prioritisation of risks within an enterprise and ability to minimize those risks by taking a specific set of actions including DP.

The TIMBUS system identifies a set of interdependent BPs from the enterprise logs, automatically detects and captures relevant context meta data, packages the collected information and provides facilities for long term preservation, monitoring and maintenance. The TIMBUS system enables the redeployment and re-execution of the partial or complete BP at a future time.

[1]http://timbusproject.net/

A high level overview of relevant research projects covering various aspects of DP, commercial and open source solutions for performing DP is presented in the next section.

1.1 Related Work

The EC has long history of supporting DP research SCAPE [20], WF4EVER [2], PROTAGE [19], APARSEN [3] and LIWA [16]. Many of the completed and currently running projects have made important contributions and are relevant to TIMBUS. SCAPE is developing a services based framework for preservation of large-scale, heterogeneous collections of complex digital objects by using semi-automated work-flows based on open source platforms. WF-4-EVER is developing a software architecture for the preservation of scientific work-flows for conducting complex research experiments in combination with relevant contexts. A smart multi-agent architecture for performing the long-term preservation of digital objects is developed in PROTAGE project. It can be integrated with existing and new preservation systems to support various aspects of a DP work-flow. A sustainable digital information infrastructure for supporting permanent access to digitally encoded information is developing based on the APARSEN framework. LIWA has developed an architecture for Web content preservation, which supports capturing content from a wide variety of sources and performs the long term interpretation of constantly evolving data archive by filtering out irrelevant information.

The following projects have some degree of overlap with DP, where the main focus lies in providing framework for the long term access to particular information resources. In the scientific domain, projects like GENESI-DR[15] and PREPARINGDARIAH[18] provide infrastructure for establishing an open digital repository for world-wide researchers to seamlessly access and share data, information and knowledge originating within different areas of science. Long term preservation in libraries and museums is investigated by ARCOMEM[4], PATHS[17], AXES[1],PAPYRUS[8] and DECIPHER[11] projects. ARCOMEM is focusing on transforming digital archives into memories structures, that can be utilised by specific community of experts, where PATHS is implementing an approach for interpreting heritage material and providing clear navigation tools for wide range of users. AXES is developing a set of tools for providing intelligent interactions with various types of digital content. An innovative ideas of understanding user queries in the context of different specific disciplines for improving underling search techniques are developed in PAPYRUS project. Using of semantic web technologies for analysing digital heritage are investigating in DECIPHER project. ENSURE[13] provides the long-term usability of data produced or controlled by commercial organisations.

Processing of digitally preserved data are explored in the following projects SOAP[21], CULTURA[7], 5-COFM[14], ETHIO-SPARE[10], ARTSENSE[5], CHESS[9] and BLOG-FOR-EVER[6]. Innovative approaches to filtering, restructuring and facilitating experience of interaction with scientific publication are developed within the scope of SOAP and CULTURA projects. Information discovery aspects in digital archives relevant to the cultural heritage domain are investigated in 5-COFM, ETHIO-SPARE and ARTSENSE projects. User engaging techniques for providing better and more efficient access to historical and cultural information as well as modern blogs are developed in CHESS and BLOG-FOR-EVER projects.

There are a number of commercial and open source products available, which address different aspects of DP. SDB[2] developed by world leading company in DP technologies provides services for storing and preserving critical digital information in reliable manner. DIAS[3], developed by IBM, addresses various aspects of long-term usability of digital information over the past decade. Rosetta developed by ExLibris[4] provides a highly scalable, secure, and easily managed DP system for preserving knowledge, libraries and other memory institution data around the world.

Alongside the commercial applications open source projects are available: Fedora-Commons, Greenstone, LOCKSS, Archivematica, DPSP, IRODS, DAITSS. CDS Invenio & CERN[5] Document Server supports preservation of articles, books, journals, photos, videos etc. and used by large number of scientific institutions worldwide. DSpace[6] developed by the Massachusetts Institute of Technology Libraries and Hewlett-Packard supports building of open digital repositories for publishing content. Eprints[7] is a set of open-source software applications for building open access services to publishing and multimedia content, which support a number of features such as meta-data extraction, access control, flexible work-flows etc.

The large verity of research, commercial and open source, in area of DP shows that the problem of DP is well-understood for data-centric information scenarios. On the other hand, scenarios where important digital information has to be preserved together with the execution contexts have been less explored. Preservation is often considered as a set of activities carried out in isolation within a single domain, without taking into account the dependencies on third-party services, information and capabilities that will be necessary to validate digital information in the future. Many existing DP solutions focus on more simple data objects which are static in nature. The unique aspect of TIMBUS is that it attempts to advance state of the art by exploring how more complex digital objects can be preserved, such as BPs with the entire execution environment. TIMBUS provides the infrastructure, which supports the user in identifying what BPs to keep, why and for how long they need to be kept.

Many modern BPs are exposed to the outside world via service oriented architectures (SOA). SOA is one of the most popular approaches, used by modern companies, for facilitating their business activities over the Internet. It provides a fast, reliable and convenient way to reach a large volume of customers world-wide. The long-term preservation of SOA based solutions is a critical necessity faced by many companies to ensure some level of business continuity. The focus on DP of BPs, where SOA is used as a framework for deliv-

[2] http://www.digital-preservation.com/
[3] http://www-935.ibm.com/services/nl/dias/
[4] http://www.exlibrisgroup.com/
[5] http://invenio-software.org/
[6] http://www.dspace.org/
[7] http://www.eprints.org/

ering services, is the main aim the TIMBUS project. Since understanding of SOA is critical for the successful design of the TIMBUS DP architecture, a high level overview is presented below.

1.2 Service Oriented Architecture

The term "service" defines a system of organised resources used for supplying specific needs to particular individuals or organisations. Services address relevant concepts from different domains [22] such as economy, business, science, etc. A typical service-based model in business [24] combines the following three service layers: *Business Service*, *E-Service* and *Web-Service*.

- *Business Service* represents the non-material equivalent of goods. They are defined as a set of activities supplied by service providers to service customers in order to deliver a specific set of values. Traditionally the majority of these services are discovered and invoked manually, while their realisation may performed by manual, semi-automated or automated fashion.
- *E-Service* is provided and executed by electronic systems in an automated fashion. IT provides an infrastructure for developing concepts such as e-service or e-commerce (electronic-commerce). Such services are executed via transactions conducted over the Internet or an internal computer network. These on-line transactions include buying and selling goods, where business is done via Electronic Data Interchange (EDI). EDI is performed by a collection of software components communicated via standardised network protocols.
- *Web Service* is the e-service consumed via Web-based protocols or Web-based programs. Separation of logical and technical layers gives a possibility of using alternative technologies for the e-service implementation. The following three types of Web-service architectures can be identified: *RPC Web Service* [23, 28, 30], *SOA Web Service* [26] and *RESTful Web Service* [29].

These services can be combined into two main business models used by service providers: *Software as a Service* (SaaS) [25] and *Internet of Services* (IoS) [24].

The SaaS Model is characterised by the following factors: provides a quicker-to-deploy strategy, quicker return of investment, frequent and automatic updates, independence from other IT components and improved usability; provides a low-risk alternative to traditionally licensed software; makes business units more focused on business transactions by eliminating dependence on supporting an IT infrastructure; facilitates a collaborative development of complex business models.

The initial focus of SaaS on the middle size companies has been expanded to the enterprise level, which changed the overall software applications market. It requires software vendors to carefully adjust their offer for meeting the constantly rising customer demand on SaaS solutions.

The IoS Model extends the concept of SaaS by providing mechanisms for discovering and invoking new services. It includes a variety of components such as standards, tools and applications for supporting business transactions. These components bring together service providers and consumers in the service-market place, where they can be more efficiently engaged in the verity of business activities.

IoS is also focusing on the creation of business networks, where elements of these networks could support SaaS models. IoS therefore provides infrastructures such as marketplace, brokerage, integration, interoperability, aggregation etc. for multiple services based on the SaaS model.

Rapid adoption of SaaS and IoS models indicates that many BPs are built on service-oriented architectures. Numerous services can be provided by different providers and operated from different geographical locations. A composition of outputs provided by each individual service is combined into particular business value, which can be utilised further by service consumers. Despite the clear advantages of SaaS and IoS, there is a danger of disappearing services and service providers (due to various reasons) by leaving some BPs partially incomplete. Considering that business continuity is not only a company desirable requirement, but also frequently a legal obligation, DP of BPs becomes as important factor of the modern business strategy. Service providers will ultimately be responsible for incorporating TIMBUS-like preservation solutions into future offerings, to support the long-term sustainability of their business models. The TIMBUS architecture presented in the next section provides the complete solution for preservation of complex BPs.

2. ARCHITECTURE

A high-level view of the TIMBUS DP architecture is shown in Figure 1. It consists of five modules: *DP Agent*, *DP Acquisition*, *Intelligent Enterprise Risk Management (IERM)*, *Legality Life-cycle Management (LLM)* and *DP Engine*.

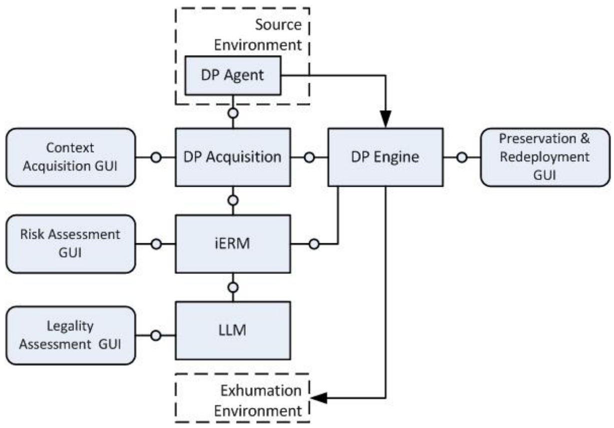

Figure 1: The high-level view of the TIMBUS DP architecture.

The *DP Agent* Module is running within the *Source* environment and capturing resources required for performing DP. The collected resources are utilised by the *DP Acquisition* Module, which extracts relevant contexts and combines them into a *Context Model Instance* (CMI). This model is annotated in *IERM* and *LLM* Modules with different risk factors and assessed according to the specified annotations. A generated risk assessment report is used by the *DP Engine* Module for selecting the most suitable preservation strategy. The selected strategy is translated into a preservation plan, which includes the complete set of instruction for execution process. During a redeployment phase the *DP Engine* Module identifies a difference between preserved and currently available environments. The identified difference is used for planing and execution by a redeployment process. All steps of preservation and redeployment processes can also be verified by a specific subset of components integrated into the *DP Engine* Module.

The DP system is integrated with *Source* and *Redeployment* environments. *Source* environment is a combination of all IT and non-IT related resources, which support the execution of BPs and need to be fully or partially preserved according to identified risk factors. The source environment consists of all infrastructure and software components required at run-time. Context information is also required for future usability, including but not limited to dependencies between business process components and the business process itself.*Redeployment* environment is a combination of IT and non-IT related resources forming an infrastructure which supports the execution of archived BPs, which can be fully or partially redeployed based on information stored within the *DP System Archive*.

Preservation and redeployment processes are controlled via four graphical user interfaces (GUI): *Contexts Acquisition*, *Risk Assessment*, *Legality Assessment* and *Preservation & Redeployment*. The *Contexts Acquisition* GUI controls processes of contexts mining and creation of CMI. The *Risk Assessment* and *Legality Assessment* GUIs control the annotation of CMI and risk impact assessment processes. *Preservation & Redeployment* defines the core set of tools for controlling planning, execution and validation processes.

2.1 DP Agent Module

The *DP Agent* Module (see Figure 2) is a combination of software components, which are running within the *Source* environment and capturing resources required for performing DP.

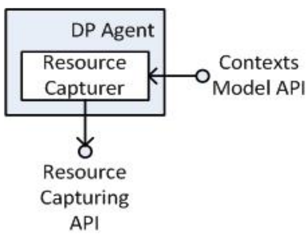

Figure 2: The architecture of the DP Agent Module.

The agent's key component *Resource Capturer* has a transparent, plug-in type architecture, which allows connecting to different resource capturing components for various IT systems. There are four main plug-in components included into the initial design: *Static Dependency Capturer*, *Dynamic Dependencies Capturer*, *Contexts Capturer* and *Event-logs Capturer*.

These plug-in components collect the key set of data required for carrying out further steps of DP. Obtained data are packaged before transferring to the DP Acquisition Module. They can be archived and/or encrypted to provide the required level of transferring efficiency and security. The created data package is transferred to the *DP Acquisition* Module via the secure peer-to-peer connection.

Control of *DP Agents* is performed via the secure communication channel, which is used for exchanging control messages. It allows control of the resource capturing process on multiple instances of the *DP Agents* from the single instance of the *DP Acquisition* Module. Such approach significantly simplifies a deployment process and minimises a cost of managing the large scale IT systems.

2.2 DP Data Acquisition

The *DP Acquisition* Module (see Figure 3) is a combination of software components, which are used for collecting and combining dependencies, contexts and event-logs from different *DP Agents* into the CMI. It consists of three software components: *Contexts Miner*, *Contexts Monitor* and *Model Weaver*.

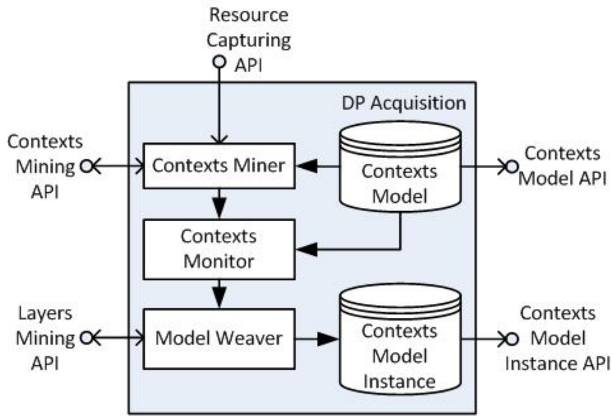

Figure 3: The architecture of the DP Acquisition Module.

The *Contexts Miner* collects data from different *DP Agents* and preforms extraction of different types of information defined by the *Context Model* (CM). At the high-level of abstraction the mining process can be split into three parts: dependencies extraction, contexts extraction and BPs extraction. The dependency extractor identifies a set of software and hardware components operating within an IT landscape and establishes relations between them. The contexts extractor discovers relevant context information relevant to the BPs, which need to be taken into consideration during the DP process. The BPs extractor analyses enterprise event-logs and extracts distinct BPs and their interrelations.

The *Contexts Monitor* receives data discovered during the mining step and performs their evaluation. The main purpose of this evaluation is to establish whether or not discovered resources contain a critical amount of data required to initiate regeneration of CMI. For example, lets assume a monitoring database hosting customer data. If a sales team member introduces a new customer record, it is not going to be considered as the critical event requires regeneration of CMI. However, if a member of the IT supporting team modifies the database design, then it may cause a significant impact on business logic regeneration of CMI for capturing introduced changes.

The *Model Weaver* combines discovered dependencies, relevant contexts and BPs into a single CMI. A few representational schemes for the CM were analysed within the TIMBUS Project. *Web Ontology Language*[8] (OWL) was selected as the most suitable model for representing the unified view on collected resources and BPs. The OWL standard combines different consecutive approaches from the Semantic Web community. OWL provides a more expressive way to define relation mappings between resources and BPs discovered within the TIMBUS project than any other relevant model. Resources identified during the mining process are represented in OWL by entities. Entities can be further sub-divided into classes and instances. Class represents an abstraction, which combines instances with the common type. Each class aggregate is a common subset of properties shared between encapsulated instances. Decisions of subdividing instances are carried out during a model designing phase. Relations between entities are labelled by descriptive terms, which allow to form meaningful connections between two or more element and perform reasoning queries.

As a result of the data acquisition operations generated, CMI contains all necessary components for performing risk assessment carried out by the *IERM* Module, which is described in the next section.

2.3 Intelligent Enterprise Risk Management Module

The *IERM* Module (see Figure 4) is a combination of software components, which are used for assessing risks associated with BPs and dependent resources. This module generates a report describing risk levels and cost values associated with failure of particular subset of BPs. It consists of four software components: *Risk Model Builder*, *Risk Annotator*, *Risk Impact Assessment* and *Risk Monitor*.

The *Risk Model Builder* allows an expert to populate *Risk Model* with relevant risk factors to a specific subset of BPs. For instance, if a particular business is heavily dependent on consumption of natural gas, the relevant risk factors will include financial losses due to supply shortages, fluctuation of market prices due to political and economical situation, natural disaster etc. When relevant risk factors are defined, the *Risk Annotator* allows the user to assign them to BP instances defined CMI. This process is performed in a semi-automatic fashion, where IERM prompts the most suitable risk factors for particular BP instance and then an expert accepts or modifies the proposed suggestion. Annotated CMI components are formed into the *Unified Risk Model* (URM). URM is a sub-model of CMI, which is only focused on supporting simulation operations. The *Risk Impact Assessment* tool uses URM to assess the impact of different risk factors on BPs, business objectives and Key Performance Indicators (KPIs). The risk impact assessment is performed by constructing a Petri Net Model[27] and running simulation

[8]http://www.w3.org/TR/owl-features/

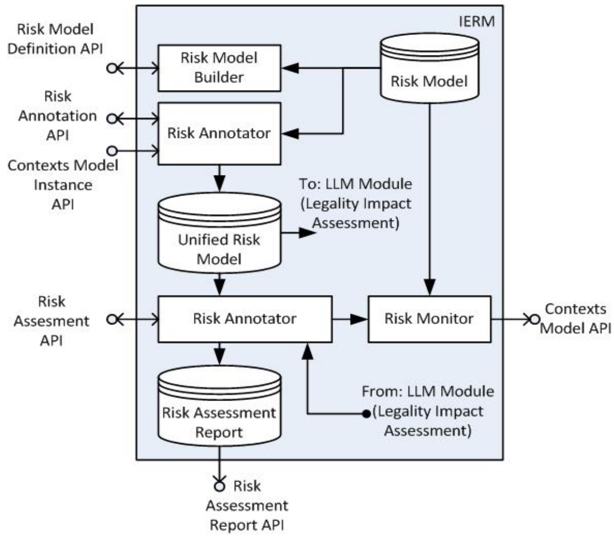

Figure 4: The architecture of the DP IERM Module.

processes. Result of these simulations are analysed to find weak points in examined BP models and compiled into the *Risk Assessment Report* (RAR).

The *Risk Monitor* constantly monitors the *DP System Archive* and CMI. It tries to detect any changes within the monitoring environment, which may lead to appearance of risk events. If the risk event is detected the monitor triggers the simulation process carried out by the *Risk Impact Assessment* and *Legality Impact Assessment* tools. The legality impact assessment is performed by the *LLM*, which is presented in the next section.

2.4 Legality Life-cycle Management Module

The *LLM* Module (see Figure 5) is a combination of software components, which are used for assessing impacts of legal issues on BPs. This module consists of two software components: *Legalities Annotator* and *Legality Impact Assessment*.

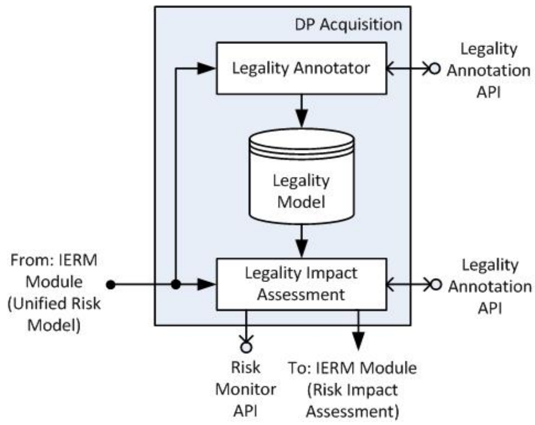

Figure 5: The architecture of the DP LLM Module.

The *Legalities Annotator* allows an expert to define legal and contractual issues relevant to a particular organisation or project and check whether or not relevant resources and BPs from the URM are compliant with these issues. Considering the high complexity of this task the majority of annotation operations are supervised by an expert. Annotated URM elements are placed into the *Legalities Store*, which is created to support the legality impact modelling process.

When the *Legalities Store* is populated the *Legalities Impact Assessment* tool checks whether or not stored legality rules are enforced for the discovered set of BPs defined in URM. Results of this assessment are sent back to the *Risk Impact Assessment* tool in the IERM module, which takes the calculated legal risk into consideration and uses it in conjunction with other risk factors.

The RAR created by *IERM* and *LLM* contains important information about BPs and various level of risks and cost associated with their failure. This information is utilised by *DP Engine* described in the next section, which identifies the most suitable preservation and redeployment strategy and performs its execution.

2.5 DP Engine

The *DP Engine* Module is a combination of software components, which are used for generating preservation and redeployment plans by utilising the RAR and CMI. This module also provides mechanisms for verification and testing different stages of preservation and redeployment processes. The *DP Engine* consists of three distinct cycles: *Preservation*, *Redeployment* and *Verification & Feedback*.

2.5.1 Preservation Cycle

The preservations cycle (see Figure 6) includes elements for preparing the preservation plan and performing its execution. It consists of the following elements: *Preservation Alternatives Assessment*, *Preservation Execution Planner*, *Process Preservation Executor*, *Preservation Monitor* and *DP System Archive*.

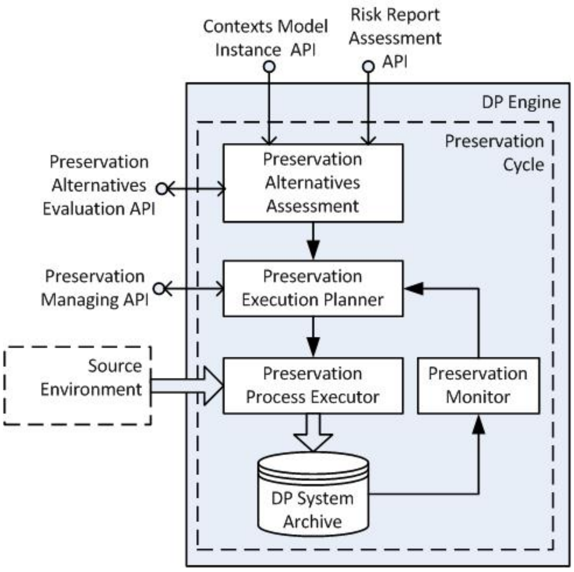

Figure 6: The architecture of the DP Engine Module for Preservation Cycle.

The *Preservation Alternatives Assessment* component analyses the generated RAR and CMI to identify the most suitable preservation strategy for most critical BPs. The assessment process takes into consideration risks associated with each BP and identifies the most effective preservation strategy. The strategy selection is performed in collaboration with an expert in an interactive mode, where the system provides different preservation alternatives and gives the expert priority to make the final decision. The selected strategy is analysed and converted into a list of instructions by the *Preservation Execution Planner*. These instructions can be executed in manual, semi-automatic or automatic fashion depending on complexity of the underlining processes. Execution of prepared instructions is carried out in the *Process Preservation Executor*. If an interaction is related to preservation of a specific IT entity, then the executor performs the automatic extraction of the requested components form the underlying IT landscape. It is followed by a set of transformation and packaging operations, required for long-term storing of selected IT entities in the *DP System Archive*. All processes are closely monitored by *Preservation Monitor*. It detects any deviations from the original script and notifies the *Preservation Execution Planner*, which initiates a re-planing phase. All decisions made during the preservation cycle are logged together with preservation steps in the specific log-containers within the *DP System Archive*.

2.5.2 Redeployment Cycle

The Redeployment cycle (see Figure 7) includes elements for preparing the redeployment plan and performing its execution. It consists of the following elements: *Redeployment Alternative Assessment*, *Redeployment Execution Planner*, *Process Redeployment Executor*, *Redeployment Monitor* and *DP System Archive*.

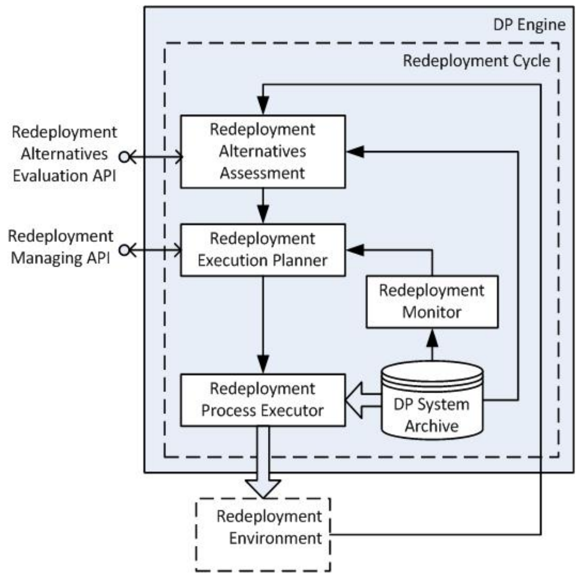

Figure 7: The architecture of the DP Engine Module for Redeployment Cycle.

The redeployment cycle performs the opposite operations to preservation. The *Redeployment Alternative Assessment* collects the information about IT landscapes, one which is initially preserved and another which is currently running within the *Redeployment* environment. Base on the collected information it identifies missing components in the currently running environment and performs assessment and selection of the best redeployment strategy. The selected strategy is utilised by the *Redeployment Execution Planner*. It generates a list of actions which exhume the selected IT components to the redeployment environment. It is important to mention that not all steps in the redeployment plan are IT related. Some of them may required verification of particular legal issues, installing the specific hardware equipment etc. The prepared redeployment plan is passed to *Process Redeployment Executor* which works in a semi-automatic fashion. It automatically executes steps for redeploying IT entities from the *DP System Archive* to the *redeployment* environment. For all other steps, which cannot be completed without an external input, it provides an assisting interface, which guides execution of these steps. All redeployment processes are closely monitored by *Redeployment Monitor*. It detects any deviations from the original script and notifies the *Redeployment Execution Planner*, which initiates a re-planing phase. All decisions are made during the redeployment cycle are logged together with redeployment steps in the specific log-containers within the *DP System Archive*.

2.5.3 Validation and Feedback Cycle

The Validation and Feedback cycle (see Figure 8) includes elements for verification and testing the preservation redeployment processes. It consists of the following elements: *Preservation Log Gap Detector* and *Verification & Feedback*.

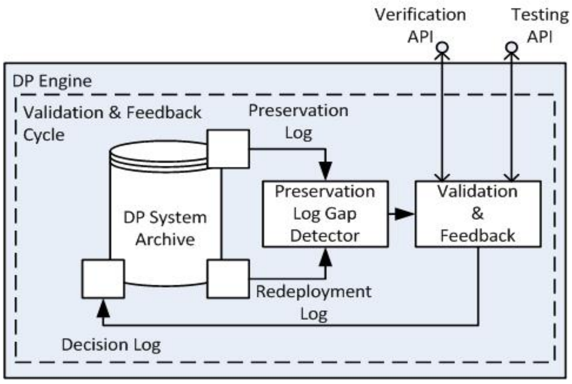

Figure 8: The architecture of the DP Engine Module for Validation & Feedback Cycle.

The *Preservation Log Gap Detector* compares the preservation and redeployment logs on potential inconsistencies. These inconsistencies may occur due to a variety of different reasons such as failures with hardware or software components during preservation or redeployment phases, unresolved legal issues, execution failures of manual steps etc. The collected information about potential issues is supplied to the *Verification & Feedback* component, which generates a set of instructions for resolving occurred issues. Considering the high complexity of this system, all operations in this phase are supervised by an expert. Another important function supported by this cycle is testing and rerunning of preserved BPs. It allows a continuous evaluation of preserved BPs and detects potential problems with the redeployment process in advanced.

3. CONCLUSIONS

The proposed TIMBUS DP Architecture provides a complete framework for preserving BPs with all relevant dependencies and contexts. It has a generic, transparent and extendable design, which overcomes limitations in existing DP solutions. It is fully compliant with all requirements and use-case scenarios developing within the TIMBUS project. These use-case scenarios cover preservation of complex BPs in commercial and scientific domains, where such aspects as initial data acquisition and processing, discovery of BPs with all relevant dependencies, enterprise and life-cycle risk management, selection and execution of the best preservation and redeployment strategies play an important part in providing long-term sustainability of business solutions.

The proposed architecture represents the complete solution for performing preservation of modern BPs and utilises the latest innovations in the area of digital preservation and combines the unique set of knowledge and expertise of all TIMBUS partners.

4. ACKNOWLEDGEMENTS

This publication would not have been possible without the support of TIMBUS project members. The authors wishes to express their gratitude to all TIMBUS members who were abundantly helpful and offered invaluable assistance, support and guidance.

5. REFERENCES

[1] Access to audiovisual archives. (AXES); CT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 269980, Retrieved April 18, 2012, from FP7-ICT Website: `http://cordis.europa.eu/projects/rcn/97842_en.html`.

[2] Advanced workflow preservation technologies for enhanced science. (WF4EVER); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270192, Retrieved April 18, 2012, from FP7-ICT Website: `http://cordis.europa.eu/projects/rcn/97462_en.html`.

[3] Alliance permanent access to the records of science in europe network. (APARSEN); ICT-2007.4.1 Digital libraries and technology-enhanced learning, Project reference: 269977, Retrieved April 18, 2012, from FP7-ICT Website: `http://cordis.europa.eu/projects/rcn/97472_en.html`.

[4] Archive communities memories. (ARCOMEM); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270239, Retrieved April 18, 2012, from FP7-ICT Website: `http://cordis.europa.eu/projects/rcn/97303_en.html`.

[5] Augumented reality supported adaptive and personalised experience in a museum based on processing real-time sensor events. (CULTURA); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270318, Retrieved April 18, 2012, from FP7-ICT Website: `http://cordis.europa.eu/projects/rcn/97475_en.html`.

[6] Blogforever. (BLOGFOREVER); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference:

269963, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/98063_en.html.

[7] Cultivating understanding and research through adaptivity. (CULTURA); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 269973, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/97304_en.html.

[8] Cultural and historical digital libraries dynamically mined from news archives. (PAPYRUS); ICT-2007.4.1 Digital libraries and technology-enhanced learning, Project reference: 215874, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/85544_en.html.

[9] Cultural heritage experiences through socio-personal interactions and storytelling. (CHESS); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270198, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/97182_en.html.

[10] Cultural heritage of christian ethiopia: Salvation, preservation and research. (ETHIO-SPARE); ERC-SG-SH5 ERC Starting Grant - Cultures and cultural production, Project reference: 240720, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/92358_en.html.

[11] Digital environment for cultural interfaces; promoting heritage, education and research. (DECIPHER); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270001, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/97302_en.html.

[12] Digital preservation for timeless business processes and services. (TIMBUS); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 269940, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/99180_en.html.

[13] Enabling knowledge sustainability usability and recovery for economic value. (ENSURE); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270000, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/98002_en.html.

[14] Five centuries of marriages. (5COFM); ERC-AG-SH6 ERC Advanced Grant - The study of the human past, Project reference: 269796, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/98760_en.html.

[15] Ground european network for earth science interoperations digital repositories. (GENESI-DR); INFRA-2007-1.2.1 Scientific Digital Repositories, Project reference: 212073, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/92602_en.html.

[16] Living web archives. (LIWA); ICT-2007.4.1 Digital libraries and technology-enhanced learning, Project reference: 216267, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/85330_en.html.

[17] Personalised access to cultural heritage spaces. (PATHS); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270082, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/97476_en.html.

[18] Preparing for the construction of the digital research infrastructure for the arts and humanities. (PREPARINGDARIAH); INFRA-2007-2.2-01 Preparatory phase for the projects in the 2006 ESFRI Roadmap, Project reference: 211583, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/88504_en.html.

[19] Preservation organizations using tools in agent environments. (PROTAGE); ICT-2007.4.1 Digital libraries and technology-enhanced learning, Project reference: 216746, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/85354_en.html.

[20] Scalable preservation environments. (SCAPE); ICT-2009.4.1 Digital Libraries and Digital Preservation, Project reference: 270137, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/97458_en.html.

[21] Study of open access publishing. (SOAP); SiS-2008-1.3.1.1 Coordination and support actions on the scientific publishing system in connection with research excellence and dissemination and sharing of knowledge , Project reference: 230220, Retrieved April 18, 2012, from FP7-ICT Website: http://cordis.europa.eu/projects/rcn/91049_en.html.

[22] G. J. Baida, Z. and B. Omelayenko. A shared service terminology for online service provisioning. In *In Proceedings of the 6th international conference on Electronic commerce (ICEC '04)*, pages 1–10. ACM, New York, NY, USA, 2004.

[23] B. Birrell, A.and Nelson. Implementing remote procedure calls. *ACM Transactions*, 2(1):39–59, 1984.

[24] V. K. Cardoso, J. and M. Winkler. Service engineering for the internet of services. *Enterprise Information Systems*, (19):15–27, 2009.

[25] R. Desisto and B. Pring. Essential saas overview and 2011 guide to saas research. *Gartner*, pages 1–16, 2011.

[26] T. Erl. Service-oriented architecture: Concepts, technology, and design. *Prentice Hall PTR, Upper Saddle River, NJ, USA.*, 2005.

[27] S. Kounev and A. Buchmann. *Vedran Kordic (ed.) Petri Net, Theory and Application*, chapter On the Use of Queueing Petri Nets for Modeling and Performance Analysis of Distributed Systems. Advanced Robotic Systems International, I-Tech Education and Publishing, Vienna, Austria, FEB 2007.

[28] T. Mowbray and W. Ruh. Inside corba - distributed object standards and applications. *Addison-Wesley-Longman*, I-XXIII:1–376, 1988.

[29] L. Richardson and S. Ruby. Restful web services. *(First ed.). O'Reilly*, 2007.

[30] J. Taylor. From p2p to web services and grids. *Springer, London*, 2005.