Towards a Decision Support Architecture for Digital Preservation of Business Processes

Martin Alexander Neumann
KIT, TECO
Karlsruhe, Germany
mneumann@teco.edu

Hossein Miri
KIT, TECO
Karlsruhe, Germany
miri@teco.edu

John Thomson
Caixa Magica Software
Lisbon, Portugal
john.thomson@caixamagica.pt

Goncalo Antunes
INESC ID
Lisbon, Portugal
goncalo.antunes@ist.utl.pt

Rudolf Mayer
Secure Business Austria
Vienna, Austria
mayer@sba-research.at

Michael Beigl
KIT, TECO
Karlsruhe, Germany
beigl@teco.edu

ABSTRACT

In this paper, we present and address a number of challenges in digital preservation of entire business processes: (1) identifying digital objects a business process depends on ("What to preserve and why?"); (2) identifying significant changes in digital objects ("When to preserve and why?"); (3) determining a re-deployment setting ("What to re-deploy and why?"). After highlighting these challenges, we illustrate some aspects of business processes that are relevant in the context of digital preservation and provide a model to capture their semantics formally. We, then, proceed to present a decision support architecture to address the challenges using the developed model. We, finally, conclude the paper by discussing the applicability of our proposed model and its associated techniques.

Keywords

Digital Preservation, Decision Support, Business Processes

1. INTRODUCTION

Digital preservation research is concerned with providing long-term *access to* and *intelligibility of digital objects*, regardless of their complexity. It focuses on preserving digital objects along with their meta-data (or contextual information) required to achieve this goal [10]. In the past, the digital preservation research has been concerned about digital objects which are static in nature, meaning they do not perform active behaviour[1] over time. In digital preservation communities, such as libraries, archives, and museums, this includes text and multimedia documents. Notably, digital objects are generated and interpreted using computational environments [9].

Recent digital preservation research activities have focused on extending established preservation approaches to dynamic digital objects; referring to those that actively perform behaviour over time. Examples of such dynamic digital objects are video games[14], interactive art[21, 2] and computational environments, such as computational scientific workflows[26]. Furthermore, an increasing amount of static digital objects are being replaced by dynamically *generated* ones—e.g. dynamic websites, results of e-science experiments, generated meta-data, etc. This content is generated using processes (i.e. computational environments) such as the simplified *documents classification process* depicted in Figure 1. This means that in order to preserve digital objects in general, the processes that define the context, within which objects are accessed and interpreted, have to be preserved as well.

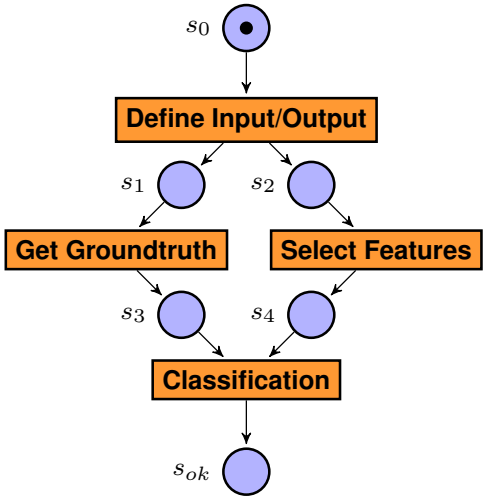

Figure 1: Classification Process to be Preserved

To provide long-term continuity in business, we are interested in digital preservation of business processes [12]. But modern business processes form considerably more complex dynamic ecosystems. A process may span many involved legal parties, is interacted with by many people having varying roles, concerns, responsibilities and authorizations, and is supported by a complex distributed service infrastructure. We, therefore, present and address here a number of challenges for the digital preservation of entire business processes that have been identified in a current digital preservation project. The project focuses on time-resilient business processes, and defines

[1] *Active behaviour* describes any externally-visible actions performed by the digital object to interact with its environment. It also refers to any actions performed purely internally which are not externally visible.

the procedures for the preservation of whole business processes as: (a) preservation planning, (b) preservation execution, and (c) preservation re-deployment (also referred to as exhumation of a preserved process). In the context of these procedures, the relevant challenges include:

What to preserve and why? During preservation planning, we identify *what* digital objects a business process depends on and *why*.

When to preserve and why? During preservation planning, we identify the relevant differences in digital objects to determine *when* to preserve a business process and *why*.

What to re-deploy and why? Before re-deployment, we identify *what* are suitable re-deployment settings, in terms of *what* preserved digital objects will be re-deployed in *what* re-deployment environments[2] and *why*.

In Section 1.1, we discuss the context of business processes relevant to digital preservation and how to model it. In Section 1.2 we discuss how to establish decision support for digital preservation activities based on these models.

In Section 1.3, we point out three reasoning tasks in the context of preservation planning, execution and re-deployment for business processes. In order to define the scope of this paper, we only focus on these three tasks (which are closely related and involve the entire preservation process). Section 2 illustrates the proposed model that has been developed for the digital preservation of business processes (which will be further revised in future). This model captures knowledge which is generally relevant to digital preservation of business processes, based on a set of representative use-cases and an enterprise modelling framework.

In Sections 3 and 4, we explain how we address the reasoning tasks based on our model and a proposed decision support architecture. We also analyze the computational complexity of our three proposed approaches. Finally, we discuss the applicability of our approach to preservation of business processes, and conclude in Section 5.

1.1 Relevant Context of Business Processes

We argue that there are many aspects in the context of a business process that have to be taken into account during preservation planning and execution, to ensure successful re-deployment of that process. We consider *successful re-deployment* as the ability to re-run a preserved process which behaves in the same way as the original one[3] [1]. Additionally, we argue that, in the context of a business process, (1) there are abstract (coarse-granular) aspects which are relevant to the entire domain of process preservation, and (2) there are more specific aspects (fine-granular) which are relevant to sub-domains of process preservation, e.g. the *class of scientific processes* or an *individual scientific experiment*, which may identify further relevant aspects. For example, at the most coarse-granular level, we have identified the following abstract categories of aspects as being relevant to the entire domain of business process preservation: (1) processes, (2) preservation requirements, (3) services, (4) software, (5) hardware, (6) data (7) licenses, (8) authorizations, and (9) people. The elements of these categories combine to form a complex inter-dependent network of different types of classes, individuals, relations and rules—they form an upper ontology capturing the knowledge relevant to business process preservation in general. This ontology may be lowered to sub-domain- or even process-specific ones to capture the knowledge relevant to the respective sub-domain.

In terms of decision support for preservation activities, there is an issue of these aspects forming large networks. Conceptually, we can use these networks of aspects to assist us in drawing conclusions from them, as illustrated in Section 1.3. However, the networks' complexities could hinder digital preservation engineers from sketching them on a blackboard and manually drawing conclusions. If we model these aspects and their inter-relations semantically adequately, we can support planning, execution and re-deployment activities using reasoning on these models. *Semantically adequately modelled* means that the model captures the semantics of the business process and its context in such a way that is suitable for automatically drawing conclusions of practical use for process preservation. The practical suitability of our model and results derived by reasoning on it have to be experimentally evaluated.

In this paper, the *context relevant to digital preservation* refers to any information that a designated user community requires to comprehend the preserved digital objects properly—i.e. intelligibility of digital objects to a designated group of people at some future point in time [10]. There are several models in the literature that capture information on context relevant to the digital preservation of digital objects. According to the Open Archival Information Systems (OAIS) Reference Model, this information is separated into *representation information* (structure and semantic information, and a representation network) and *preservation description information* (reference, context, provenance, and fixity information) [17].

Examples of models (and related formats) are: (1) the METS and OAI-ORE formats for packaging and exchanging of digital resources; (2) FRBR[16] to model information realization and versioning problems in libraries; (3) Dublin Core, MODS and MARC to record bibliographic information; (4) the ABC Ontology[19], the Open Provenance Model[22], the PROV Data Model[4], the SHAMAN Context Model[6], and the PREMIS Data Dictionary for Preservation Metadata[25] capture provenance information to model life-cycles of digital objects in and/or outside of digital archives; (5) CIDOC CRM[22] to integrate heterogeneous cultural heritage information; and (6) representation information networks[17] to structure representation information.

These models provide means for modelling OAIS-relevant information on digital objects with different focus and varying levels of detail. They are concerned about structural generic semantic aspects of digital objects, and about processes in the context of digital objects. But they do not yet characterize "behavioural aspects" of (dynamic) digital objects themselves. In addition, they do not yet focus on semantic aspects specifically relevant in the context of business processes or workflows.

[2] An adjustable part of a re-deployment environment may be adapted during the re-deployment procedure to accommodate for the process-specific situation established by the preserved digital objects and parts of a re-deployment environment which are fixed.

[3] It behaves equivalent according to an equivalence notion, such as trace equivalence[24], and equivalent in terms of relevant modalities, such as causality and time. Both aspects are determined by the requirements of process preservation in general, but also by the requirements of preserving the process in focus.

From our perspective, executional aspects are relevant, because we have to model systems which are complex objects on the one hand (as business processes have a compositional structure of inter-related parts), and those which perform actions (behave) on the other hand. Thus, in addition to a structural and generic semantics notion and model, we need a notion and model of behaviour which is adequately applicable to digital preservation of business processes. As stated before, this notion and model of behaviour has to accomplish the above goal of enabling successful re-deployment of a preserved process. As a consequence, we extend the interpretation of the term *digital preservation relevant context* to: information that a designated user community requires to comprehend archived digital objects properly, as well as information that a designated user community requires to verify the execution of a re-deployed behavioural system. We also propose a novel modelling approach for the digital preservation of business processes that captures relevant structural, semantic, and behavioural aspects, to enable successful re-deployment of a preserved process. However, as mentioned above, whether the modelling approach achieves this goal has yet to be evaluated in representative case studies of whole process preservation.

To foster preservation of computational scientific workflows, models for context and behaviour of such processes are proposed in [26, 13]. Context is modelled as sets of required services and data in [26], and [13] proposes a notion of process behaviour which seems equivalent to condition-event structures (which are revisited in Section 2 and we promote too since [23]). To build on this research, in this paper, we extend our notion of process behaviour by time and propose a flexible context modelling approach.

1.2 Decision Support for Digital Preservation

As mentioned before, the introduced models used for capturing the context relevant to digital objects focus on their respective domains which they model to a certain level of detail and at a certain level formality. From a knowledge representation perspective, they all are based on individual domain-specific ontologies; i.e. in general, to model digital objects and information about their context, the ontologies provide relevant: (1) classes, (2) instances of these classes, (3) relations between these classes and instances, and (4) additional rule-like statements on classes, individuals and relations.

Enabling tractable automated reasoning on these models requires them to be based on an adequately expressive and decidable language which sound and complete inference mechanisms can operate on. This provides the ability to provide explainable and correct answers to any expressible decision problem or query on the models in feasible time. The required level of formality is provided by some of the covered models. For example, the Open Provenance Model, the PROV Data Model, the ABC Ontology, the PREMIS Data Dictionary, and CIDOC CRM have been implement in the Web Ontology Language 2 DL (OWL 2 DL)[15] (or subsumed language fragments).

Besides capturing behavioural aspects, our modelling approach captures the introduced structural and semantic aspects. Both are modelled on a "semantically rich" (i.e. formal and detailed) level, based on an ontology language in general. This has two advantages: (1) automated reasoners that assist during preservation planning, execution, and re-deployment can directly operate on the knowledge maintained along with a preserved digital object; and (2) the knowledge kept with a preserved digital object can even be specific to this object, which means that the model is specific to the preserved business process. A reasoner would, then, directly be able to draw conclusions from it without having to combine the knowledge kept with the digital object with the background knowledge kept inside the reasoner itself. Combining both would be necessary, if the reasoner would bring in some knowledge in addition to the knowledge kept with a digital object. In this case, both knowledge bases are in danger of contradicting each other and, therefore, hard to combine [7]—in particular, if both knowledge bases originate from different contexts, such as points in time or user communities. This implies another positive of our approach: in general, reasoners do not have to be sub-domain- or process-specifically adapted and are thus time-resilient.

As already mentioned, we promote the use of an ontology to model the information and knowledge on digital objects, and also to design object-specific models to accommodate for specific digital preservation requirements of an object. For example, in one scenario it might be sufficient for re-deployment of a business process if the requirements stipulate causal trace-equivalent behaviour after re-deployment. However, in the case of a scientific experiment, causality and exact timing are likely to be very relevant. Therefore, if we would like to assist preservation planning in answering the question "what to preserve and why?" for both processes, there is no generic strategy to answer it. For the first process, it would be sufficient to only preserve technical requirements down to the operating systems which in this example are known to provide a runtime environment that preserves causality. In the case of the second process, we might need to preserve technical requirements down to the hardware, which is assumed to provide cycle-time accurate timing. Therefore here, we need two different strategies (or policies) to determine which parts of the business processes are required to be captured. As the strategy is specific to the digital object in focus, it must be kept with the object itself and not the reasoner.

We envision that many digital preservation related questions are specific to digital objects, analogous to the illustrated example. Answering these digital preservation questions depends on the context (or situation). Therefore, we argue that it is important to provide the ability to capture object-specific knowledge for their digital preservation, in particular for business processes. This would improve the understanding of preserved digital objects without the need for background knowledge, and also enable generic reasoning mechanisms to act on the preserved digital object only, to assist in preservation activities, such as planning, execution, and re-deployment.

The digital preservation research has already implemented decision support approaches. The most recent one is Plato[3]. In contrast to our methodology, Plato focuses on digital objects which are static in nature, and as such do not perform active behaviour over time; e.g. text documents and images. Plato provides a reasoning framework for identifying relevant actions to preserve a digital object. In general, this idea complements the approach pursued in this paper, as we do not discuss the question of "how to preserve a digital object?". And, as we are concerned about dynamic digital objects, Plato's applicability to this domain is a relevant future aspect.

To achieve its goal, Plato (1) defines generic features of digital objects, such as the presence of intellectual property rights issues; (2) defines more specific features of classes of digital objects, such as compression characteristics of image formats; (3) devises methods to extract these features from digital objects, such as by using tools or performing manual experiments; and (4) proposes a method to conclude optimal preservation actions from the features of a

digital object. This methodology is in line with our vision and requirement of being able to draw conclusions from the model of a digital object only. To provide this, a generic mechanism is proposed that calculates and compares the *utilities* of preservation actions on a unified scale, whereby the feature extraction techniques of a digital object are responsible for providing a strategy to map their outputs onto this scale.

1.3 Process Preservation Challenges

In order to be correctly rendered, a digital object needs a technological context resulting from the combination of specific hardware and software. Moreover, in order to be correctly understood by humans, the organizational, business, and social contexts surrounding the object are also needed. The Digital Preservation Europe Research Roadmap, published in 2007, defines the context of a digital object as the "representation of known properties associated with and the operations that have been carried out on it"[11]. On the one hand, these properties might include information about the technology used, but on the other hand they might consist of legal requirements, existing knowledge, and user requirements. The operations performed on an object might even include the processes that originated the object itself.

The determination of the relevant context of a digital object becomes even more challenging if complex digital objects such as workflow or business process specifications are considered. Those types of objects are dependent on a highly complex and distributed technical infrastructure hosted in complex and diverse organizational settings, sometimes involving multiple organizations. This creates a complex dependency network involving the object and other complex objects on which its correct rendering and understanding depends. However, not all context might be relevant for being able to preserve and successfully re-deploy a process in the future. Some of the context might not even be available at all—for example, if the details of some external services are not accessible. In general, a selective approach for determining the context of a process should be pursued, which enables to select the partial context which is use case-specifically required for preservation of a process. Otherwise, it might lead to resource waste, and might even cause the costs of preservation to surpass its potential benefits. In that sense, the first preservation challenge faced when dealing with the preservation of business processes is "what to preserve and why?".

After the identification of the relevant contextual information, it becomes necessary to determine how to approach the capturing and preservation of the process and relevant context. In other words, it is important to determine what preservation actions should be performed. As introduced, this issue has so far been addressed by Plato. It is assumed that surpassing this challenge will result in the successful execution of the preservation actions that will allow the process and its relevant context to be preserved.

Furthermore, as a process and its context have to be captured at a determined point in time during preservation, it becomes crucial to monitor the original process to detect any changes in process behaviour. Since those changes are potentially relevant to capture to preserve the most recently working version of a process, another preservation challenge being faced is "when to preserve and why?". Facing this challenge successfully will involve having several snapshots of a process and its relevant context documenting the main events happening during its life-cycle.

Challenges are also faced during the re-deployment of a preserved process. Since digital preservation concerns the long-term, it is highly probable that the original deployment setting is partly or not available at all. The preserved context model provides indicators to what are suitable re-deployment settings for the preserved processes. The re-deployed environment might need adaptation during the re-deployment procedure in order to re-establish any situation of interest. In general, an optimizing approach for determining re-deployment settings should be pursued, to minimize re-deployment efforts and therefore associated costs. Hence, a challenge that must also be faced in the re-deployment of business processes includes knowing "what to re-deploy and why?".

After the identification of the re-deployment setting, it becomes necessary to determine how to approach the re-deployment itself. Thus, it is crucial to determine what re-deployment actions should be performed. This issue is, again, analogous to what is already being addressed by Plato. And it will be surpassed if the re-deployment of the process and environment allow for the correct re-execution of the process. This is an issue we are trying to resolve by comparing a re-deployed process to its original process based on the comparison of the outputs produced by them, as presented in [20].

2. CONTEXT MODEL

Our *context model* describes business processes and their context, both of which are scoped to aspects relevant to the digital preservation of the processes. The context model is a formal ontology that can be instantiated, or specialized, to model individual digital preservation settings (which involve concrete business processes and their context). The instantiation of the model involves the definition of classes, individuals, relations, and statements which are specific to the digital preservation setting. This provides the ability to model processes and their digital preservation-relevant context in a semantically rich way, as motivated in Section 1. To specify our ontology and scope it to the domain of digital preservation of business processes, we have investigated which classes, individuals, relations, and logical statements apply to the entire domain of digital preservation of business processes. The design methodology (middle-out approach) and preliminary details on the contents of our ontology are presented in [20].

Furthermore, as introduced in [23], we have identified *condition-event structures* (or 1-safe petri nets) as being an adequate notion for modelling the structure and causal behaviour of business processes. It is an approach for design and *efficient* verification which clearly formulates causal behaviour of concurrent systems [8]. To be able to additionally model temporal behaviour of business processes, as required in this work, we extend our notion to *time condition-event structures*. This approach allows to model *causal and temporal behaviour* of concurrent processes for design and verification.

A model $\mathcal{M} := \langle \mathcal{B}, \mathcal{C} \rangle$ consists of a set of business processes $\mathcal{B}$ and a context $\mathcal{C}$. A condition-event structure $\mathcal{N}^{c/e} := \langle \mathcal{P}, \mathcal{T}, \mathcal{F}, m_0 \rangle$ consists of a set of *places* $\mathcal{P}$ encoding *conditions* and a set $\mathcal{T}$ of *transitions* encoding *events*, where $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is the set of edges of the net and m_0 is the *initial marking*. Here, a function $m_i : P \to \{0, 1\}$ is called a *marking*. A transition t is activated ("may fire") in a marking m_i iff for all p holds: (1) if $(t, p) \in \mathcal{F}$ then $m_i(p) = 0$, and (2) if $(p, t) \in \mathcal{F}$ then $m_i(p) = 1$. A sequence of "fired" transitions $t_i \to \ldots \to t_j$ is called a *trace*.

A time condition-event structure $\mathcal{N}^{t,c/e} := \langle \mathcal{P}, \mathcal{T}, \mathcal{F}, m_0, l \rangle$ consists of a condition-event structure $\langle \mathcal{P}, \mathcal{T}, \mathcal{F}, m_0 \rangle$ and a *time labelling function* $l : \mathcal{T} \to \mathbb{N}_{\geq 0} \times \mathbb{N}_{\geq 0} \cup \{\infty\}$ whereby for all

$t = (t_i^\circ, t_i^\bullet)$ holds: $t_i^\circ \leq t_i^\bullet$ and $t_i^\bullet < \infty$. All t_i° are called *earliest firing times* and all $t_i^\bullet$ are called *latest firing times*. A transition "may fire" the earliest at its t_i° and "has to fire" the latest at its $t_i^\bullet$ since its activation. Furthermore, $j_i : \mathcal{T} \to \mathbb{N}_{\geq 0} \cup \{\phi\}$ is a *clock function* that gives the time which has elapsed since a transition t has been activated. In consequence, for all t_i holds: $j_j(t_i) \geq t_i^\circ$ and $j_j(t_i) \leq t_i^\bullet$. A sequence of time-annotated "fired" transitions $(t_i, j_i) \to \ldots \to (t_j, j_j)$ is called a *time trace*.

Now, the set of business processes $\mathcal{B}$ in our model can be defined as a set of time condition-event structures: $\mathcal{N}_i^{t,c/e} \in \mathcal{B}$. Furthermore, the context $\mathcal{C} := \langle \mathcal{E}, \mathcal{R}, \mathcal{S} \rangle$ consists of a set of classes $\mathcal{E}$, a set of relations $\mathcal{R}$ and, a set of logical statements $\mathcal{S}$. Each class $e_i := \{i_0 \ldots i_n\}$ is a set of individuals i_j. Each relation $r_i \subseteq (\mathcal{T} \times \mathcal{E}) \cup (\mathcal{E} \times \mathcal{E})$ relates transitions (i.e. events) to classes, and classes to classes. Each logical statement s_i is a horn-formula in first-order logic[18] whereby its predicates are restricted to the relations in $\mathcal{E}$ and $\mathcal{R}$.

3. ADDRESSING THE CHALLENGES

Figure 2 presents our proposed architecture to provide decision support in terms of the highlighted challenges. In a concrete digital preservation setting, the context model (1) introduced in Section 2 is firstly fed into the *Model Builder* to create a specialized instance of the model—it ingests our ontology which is specific to the entire domain of process preservation to create an instance of it specific to the process. Secondly, to create this instance, relevant knowledge from knowledge bases[4] (2—such as data formats and software licenses) and process-specific details (3—such as process-specific preservation requirements, and involved software and hardware) are added to the ontology by the *Model Builder*. The process-specific details may either be automatically extracted from a business process (e.g. software and hardware) or manually input by digital preservation engineers (e.g. preservation requirements).

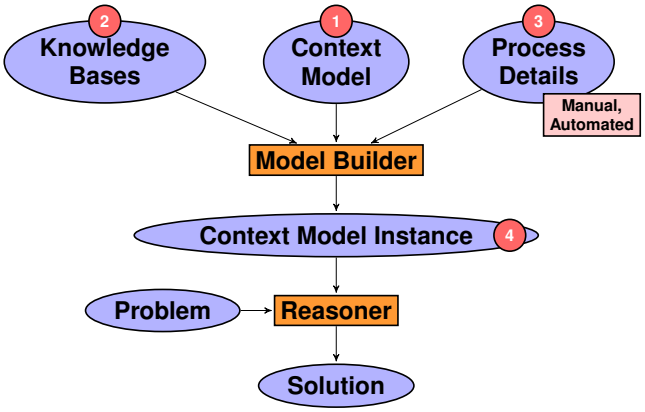

Figure 2: Decision Support Architecture

The produced model (4) captures all the knowledge relevant to the digital preservation of the process in focus and will accompany the process during its entire life-cycle in a preservation archive. Furthermore, the model contains the knowledge required to provide decision support to the three presented preservation challenges, as will be illustrated in the following sections. In general, as our model is based on individuals (objects), classes (unary relations), binary relations and horn formula in the two-variable fragment of first-order logic, the produced model can be handed over to various types of semantic reasoners (such as "off-the-shelf" description logic or firstorder logic reasoners) to conclude solutions from given problems based on the given model only.

[4]The knowledge bases conceptually are part of the ingested context model, but are kept separate from it in our implementation.

3.1 Objects to Preserve

As outlined earlier, answering the question of "what to preserve and why?" can be reduced to establishing a notion of what is *required* by a process to be preserved and successfully re-deployed. This notion is determined by preservation requirements which are relevant to the entire domain of process preservation, and more specific requirements which are relevant to sub-domains of process preservation. For example, as illustrated earlier, each process imposes individual requirements on its causality and timing equivalence. Therefore, this notion is specific to the process and the digital preservation setting[5] (called *process-specific* in the following).

In general and in our ontology, there are several ways to model *what is required* by a process. One approach is to explicitly model a *requires relation*. For example, we could say that "a program requires an operating system, which requires a machine, which requires an operator". This is a semantically rather limited notion, and there is no need for our idea of a "semantically rich" context model. But this approach does not provide a process-specific notion of what is required. If we capture a model of only *requires relations*, for example, of all software and hardware components involved in a process, we cannot tell what components are "really necessary" for successful re-deployment without inspecting the model and deleting information. This approach is likely to lose information relevant to yet unknown re-deployment settings.

Another approach would be to implicitly model a *requires relation* by declaring other relations, such as *runsOn*, *isInstalledOn*, *isOperatedBy* of being a subtype of the *requires relation*. Based on this, we could process-specifically select which relations determine *what is required*. For example, we could model that "a program isInstalledOn an operating system, which runsOn a machine, which isOperatedBy an operator" and conclude that all four individuals are required by our process. But this is still semantically rather limited, as we still could not process-specifically distinguish, for example, "really necessary" software and hardware components from "not really necessary" ones.

Therefore, we argue that a more expressive approach is required which provides a more complex notion of *what is required*, and we propose horn formula in the two-variable fragment of first-order logic to express this process-specifically in our ontology. It allows to express that all objects that satisfy a complex statement are required. For example, "it is only necessary to preserve an operating system if it is proprietarily licensed". We are in the process of implementing this approach using OWL 2 DL and the Pellet reasoner[27]. Based on this, all the problems presented to our reasoning engine are decidable, although the employed language exposes a worst-case computational complexity in reasoning of N2EXPTIME[5]. Our future efforts will determine which language fragments are required in the process preservation practice to improve on the complexity and whether it is a computationally tractable approach.

[5]In this context, the *setting* particularly refers to the temporal preservation horizon which determines setting-specific aspects such as available technologies and relevant user communities of the future.

3.2 Events to Preserve

As mentioned earlier and discussed in [23], answering the question of "when to preserve and why?" can be reduced to establishing a notion of what is the *difference* between the process now and when it has been preserved the last time. If this difference exceeds some level of relevance, then a new trigger to preservation execution is determined. Again, this notion of *what a relevant difference in what modalities is*, is process-specific, as each process imposes individual requirements on its causality and timing equivalence.

We propose a notion of *trace equivalence* to detect *relevant differences* in causality and timing behaviour of a process at two different times. Our idea is based on the detection of relevant differences in the execution traces of processes under equivalent contextual conditions (regarding their interaction with the environment, such as values of inputs). Based on the *traces* and *time traces* of processes that are defined in our model (in Section 2), we can compare traces stored in two models with each other. Comparing any two traces requires that they have been taken under equivalent contextual conditions—they are called *comparable traces* in the following. We propose a *process-agnostic* notion of difference in the qualitative order of events, and a *process-specific* notion of difference in the quantitative order of events.

Regarding the qualitative difference notion, any change in the qualitative order of events between two comparable traces marks a *relevant difference*. Regarding the quantitative difference notion, *deviations* of an event's timing (in a *time trace*) from its time interval[6] marks a trace which deviates from its process specification. Incorporating the process (of which the trace has been taken) is important in this case, as the quantitative difference notion is process-specific. Two *comparable time traces differ relevantly* from each other, if and only if one of them deviates from the timing interval specification and the other one does not. Each process defines an individual interval of expected timing values for each event, as defined in our model in Section 2. These individual interval information can be either given by expert knowledge or by profiling a process.

The causal and timing behaviour of a process, during its execution under specific contextual conditions, is given by one *time trace* in our model. If we want to capture the behaviour of a process under varying contextual conditions, we need to capture (in our model) a set of *time traces*, along with their contextual conditions. To compare two processes, we compare their trace sets. The trace sets have to have been taken under the same varying conditions. Each two traces that have been taken under the equivalent conditions have to be compared with each other. If this fails on at least one set of two traces, a *relevant difference* has been identified. When this approach is applied to monitoring of a process which is to be preserved, the identified relevant difference represents a trigger ("when to preserve and why?") to preservation of the process.

Analogously to the previous challenge presented in Section 3.1, we are in the process of implementing this approach using a tractable fragment of OWL 2 DL and the Pellet reasoner.

3.3 Objects to Re-Deploy

Although it seems analogous, answering the question of "what to re-deploy and and why?" is considerably more complex than the earlier discussed question of "what to preserve and why?". In addition to the preserved process, we have to take into consideration the environment we are going to re-deploy the process into. The re-deployment environment will consist of a *fixed* and a *flexible part*. This means that there will be an unchangeable (or constrained) part in the re-deployment environment, for example, some machines in a data center, and a changeable (or un-constrained) part of the environment, for example, the possibility of selecting an alternative operating system running on these machines in the data center. We reduce answering the question "what to re-deploy and why?" to a notion of what is *required* to re-deploy a preserved process. Again, this notion is process-specific, even more than in our previous challenges as the re-deployment environment takes a major role in our reasoning problem.

In reasoning, we have to take three instances of our context model into account, which have to be determined first: a model of the *preserved process*, a model of the *constrained environment*, and a model of the *un-constrained environment*. Afterwards, we will determine all *feasible re-deployment alternatives* and pick an *optimal one*. This is performed by identifying the *difference* between the preserved process and the constrained environment in more detail. There are four possible outcomes of this evaluation:

None The constrained environment is identical to the environment when the process has been preserved. Therefore, combining their models does not introduce inconsistencies into our ontology, and neither our process, nor the environment have to be adapted to re-deploy.

Overlap The preserved process and the constrained environment *overlap*. This means that their combined model contains overlapping sub-graphs which address the same issue, meaning which are not allowed to overlap and therefore introduce inconsistencies into the ontology. For example, two different operating systems on the machines in the data center.

Gap There is a *gap* between the preserved process and the constrained environment. This means that their combined model contain sub-graphs which are disconnected from each other although they need to be connected, meaning the disconnected sub-graphs introduce inconsistencies into the ontology too. For example, if none of the models cover operating systems.

Both The preserved process and the constrained environment partially *overlap* at one to many points and partially have one to many *gaps* between each other.

After the situation has been sorted out thoroughly, and if we have determined that we cannot immediately re-deploy, we continue in a second step to determine all feasible re-deployment alternatives. This is based on the models of the preserved process, and both environment models (constrained and un-constrained). The reasoner applies the following strategies in solving any gaps or overlaps:

Overlap In case of an overlap between the models of the preserved process and the constrained environment, the reasoner will take parts out of the model of the preserved process to find options that eliminate the inconsistency from our ontology. This may mean that the reasoner takes larger parts out of the model than the actual overlap, which are filled by parts from the model of the un-constrained environment.

[6]Refers to the time interval specification of the event in the *time condition-event structure* of the process (in our model).

Gap In case of a gap between the models, the reasoner uses the model of the un-constrained environment to find all options to fill this gap and thus eliminate the inconsistency from the model. This may even mean that the reasoner has to take out parts from the model of the preserved process.

Afterwards, all alternatives are ranked to conclude the optimal re-deployment alternative. We are in the process of implementing this reasoning procedure based on linear optimizers, specifically the APT-PBO solver[28], which allows us to determine many feasible re-deployment alternatives and rank them according to a process-specific cost function. APT-PBO is different from other similar solvers in that it acts as an interactive system and as such the proposed solutions can be navigated and further decisions taken that is likely to be important in the re-deployment scenario.

An illustrative example of a technical scenario is having a preserved software library (used by a business application) that will not work with the re-deployment environment. The library may have had a known security flaw meaning that in a re-deployment environment it would have to be updated to a version that included the security fix. Another possible issue could be that the library cannot be used because of licensing issues or doesn't work in combination with some other system that is in place in the new environment. The reasoner would then, based on the context models, try to determine feasible alternatives to the library to update it and rank them according to criteria. This procedure involves the reasoner trying to determine what else would be affected by updating the library. If other software is affected by the update, this could additionally be notified to an digital preservation engineer and then either a more updated version can be installed or a manually-proposed alternative be applied which fulfils the requirements.

4. DECISION SUPPORT WORKFLOW

In [20], we present in detail a classification process which is also sketched in Figure 1. The process builds a music genre classifier based on features extracted from given training data, and afterwards classifies given input data based on features extracted from them. Notably, the process involves a variety of free and proprietary data formats, such as HTML and MP3, and external services, e.g. for feature extraction. We have modelled its behaviour and required formats, software, hardware and licenses in a context model instance. Based on this, we informally illustrate here the application of our proposed procedure to provide decision support to the challenge of "what to preserve and why?" on this model instance.

Instantiate Context Model *The first step in the decision support workflow is to populate the context model semi-automatically using extraction tools and expert knowledge of digital preservation engineers.*

For example, we have extracted a process model from the employed workflow engine which yields the process' behaviour and its external service dependencies. Furthermore, we have extracted a directed graph of software dependencies of the workflow engine from the software package repository of the operating system.

Specify Requirements *Next, the specific requirements of our digital preservation setting to evaluate the question of "what to preserve and why?" have to be specified. This covers conditions which are sufficient to be satisfied by an individual such that it has to be preserved. And this covers conditions which are required to be satisfied by an individual such that it can be preserved.*

In this example, we follow a straight-forward approach in specifying whether an individual has to be preserved. Analogously to representation information networks[17], we specify dependencies explicitly by introducing a transitive relation called "requires" which subsumes all other relations in the context model instance. Now, we declare that "being (transitively) required by the process" is sufficient for an individual for having to be preserved.

Furthermore, we assume that we are required to preserve for at least 10 years (i.e. long-term). The knowledge modelled in the context model yields, for example, that required software must not depend on external services. Software individuals can only be preserved if they satisfy this requirement. But for our process we relax this by allowing *feature extractors* to be preservable if they exchange data in a standardized format, such as ARFF.

Specialize Context Model Now, to provide this relaxation of the digital preservation requirements, the context model has to be inspected and its classes and relations specialized to process-specific needs. At this point, the workflow becomes iteratively, as in the next step the model has to be re-instantiated to populate the specialization appropriately.

For example, we have added the concept of *feature extractors* (a specialization of external services), which is a relevant concept of our process to reason about its preservability.

Evaluate Results *And finally, our proposed reasoning engine is employed to determine (1) if these requirements can be satisfied, and (2) what sub-graphs of the context model instance satisfy them.*

Without having specified that feature extractors are preservable, our procedure would conclude that the desired long-term preservation cannot be performed—yielding the non-preservable external feature extraction service as the reason. After expert consultation, we have relaxed this requirement, which yields at least one preservable sub-graph of our context model instance.

5. CONCLUSION

In this paper, we have motivated the necessity for digital preservation research on dynamic digital objects, such as processes generating (a) dynamic websites, (b) results in e-science experiments, and (c) metadata. Based on this, we have illustrated three challenges in decision making that span the procedures linked with digital preservation of business processes (planning, execution and re-deployment). These challenges have been identified in the context of a digital preservation project that focuses on time-resilient business processes. The challenges were: (1) identifying digital objects a business process depends on; (2) identifying significant changes in those objects; and (3) determining suitable re-deployment settings. As motivated earlier, due to the complexity of the tasks at hand and its inherently associated efforts, providing techniques in solving them using decision support tools will ease the duties of involved stakeholders.

In previous work, we have already presented ideas to partially address the first and second challenges, and we have outlined their application in a case study, a scientific workflow. A context model instance is semi-automatically generated and a method for verifying the workflow's behaviour after re-deployment is presented in [20]. In this paper we have extended this work by (a) devising a procedure for determining "what to preserve and why?" from a given context model instance, and (b) by specifying an equivalence notion on *time traces* to detect relevant changes in process behaviour on a generic base. Furthermore, in [23], we propose an approach to monitoring of business processes to trigger their digital preservation and verifying their causal behaviour. Here, we have extended this notion to enable verification of causal and temporal behaviour of processes.

In addition, in this paper, we have presented an architecture to assist in the decision making of the preservation procedures in general. The architecture has been based on a knowledge representation technique specifically tailored to process preservation, called the *context model*. We have, also, presented how we are addressing the identified challenges using the architecture and reasoners applicable to our model—in general, logic-based reasoning engines (Pellet and APT-PBO) being applied. In [12] we present the integration of the model and several instances of our proposed architecture (which address the challenges) into an architecture for digital preservation of entire business processes. Our future efforts are focused on implementing and evaluating the covered modules.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the funding by the European Commission under the ICT project "TIMBUS" (Project No. 269940, FP7-ICT-2009-6) within the 7th Framework Programme.

7. REFERENCES

[1] J. Barateiro, D. Draws, M. A. Neumann, and S. Strodl. Digital Preservation Challenges on Software Life Cycle. pages 487–490, 2012.

[2] C. Becker, G. Kolar, J. Kueng, and A. Rauber. Preserving Interactive Multimedia Art: A Case Study in Preservation Planning. In *Proceedings of the 10th International Conference on Asian Digital Libraries*, ICADL'07, pages 257–266, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] C. Becker and A. Rauber. Decision criteria in digital preservation: What to measure and how. *Journal of the American Society for Information Science and Technology*, 62(6):1009–1028, June 2011.

[4] K. Belhajjame, H. Deus, D. Garijo, G. Klyne, P. Missier, S. Soiland-Reyes, and S. Zednik. *PROV Model Primer*. World Wide Web Consortium, 2012.

[5] R. Bembenik, L. Skonieczny, H. Rybiński, and M. Niezgodka. *Intelligent Tools for Building a Scientific Information Platform*. Studies in Computational Intelligence. Springer-Verlag, 2012.

[6] H. Brocks, A. Kranstedt, G. Jäschke, and M. Hemmje. Modeling Context for Digital Preservation. In *Smart Information and Knowledge Management*, volume 260 of *Studies in Computational Intelligence*, pages 197–226. Springer, 2010.

[7] A. Calì, T. Lukasiewicz, L. Predoiu, and H. Stuckenschmidt. Tightly Coupled Probabilistic Description Logic Programs for the Semantic Web. *Journal on Data Semantics*, pages 95–130, 2009.

[8] A. Cheng, J. Esparza, and J. Palsberg. Complexity results for 1-safe nets. *Theoretical Computer Science*, 147(1-2):117–136, Aug. 1995.

[9] C. Chou, A. Dappert, J. Delve, and S. Peyrard. Describing Digital Object Environments in PREMIS. In *Proceedings of the 9th International Conference on Preservation of Digital Objects*, iPRES 2012, 2012. (to appear).

[10] M. Day. Metadata for Digital Preservation: A Review of Recent Developments. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '01, pages 161–172, London, UK, UK, 2001. Springer-Verlag.

[11] DigitalPreservationEurope Partners. DPE Digital Preservation Research Roadmap. Public Deliverable D7.2, DPE, 2007.

[12] M. Galushka, P. Taylor, W. Gilani, J. Thomson, S. Strodl, and M. A. Neumann. Digital Preservation Of Business Processes with TIMBUS Architecture. In *Proceedings of the 9th International Conference on Preservation of Digital Objects*, iPRES 2012. (to appear).

[13] D. Garijo and Y. Gil. A new approach for publishing workflows: abstractions, standards, and linked data. In *Proceedings of the 6th Workshop on workflows in Support of large-scale science*, WORKS '11, pages 47–56, New York, NY, USA, 2011. ACM.

[14] M. Guttenbrunner, C. Becker, and A. Rauber. Keeping the Game Alive: Evaluating Strategies for the Preservation of Console Video Games. *International Journal of Digital Curation*, 5(1), 2010.

[15] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. *OWL 2 Web Ontology Language: Primer*. World Wide Web Consortium, 2009.

[16] International Federation of Library Associations. Functional Requirements for Bibliographic Records - Final Report. Technical report, IFLA, 1998.

[17] International Organization For Standardization. OAIS: Open Archival Information System – Reference Model. 2003. ISO 14721:2003.

[18] R. A. Kowalski. Predicate Logic as Programming Language. In *IFIP Congress*, pages 569–574, 1974.

[19] C. Lagoze and J. Hunter. The ABC Ontology and Model. In *Proceedings of the International Conference on Dublin Core and Metadata Applications*, pages 160–176. National Institute of Informatics, Tokyo, Japan, 2001.

[20] R. Mayer, A. Rauber, M. A. Neumann, J. Thomson, and G. Antunes. Preserving Scientific Processes from Design to Publications. In *Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries*, TPDL 2012. Springer-Verlag, 2012.

[21] A. McHugh, L. Konstantelos, and M. Barr. Reflections on Preserving the State of New Media Art. In *Proceedings of the Archiving Conference*, 2010.

[22] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The Open Provenance Model core specification (v1.1). *Future Gener. Comput. Syst.*, 27(6):743–756, June 2011.

[23] M. A. Neumann, T. Riedel, P. Taylor, H. R. Schmidtke, and M. Beigl. Monitoring for Digital Preservation of Processes. In *Proceedings of the 7th international and interdisciplinary conference on Modeling and using context*, CONTEXT '11, pages 214–220, Berlin, Heidelberg, 2011. Springer-Verlag.

[24] L. Pomello, G. Rozenberg, and C. Simone. A survey of equivalence notions for net based systems. In *Advances in Petri Nets 1992*, pages 410–472. Springer-Verlag, London, UK, UK, 1992.

[25] Premis Editorial Committee. Data Dictionary for Preservation Metadata: PREMIS version 2.0. (March), 2008.

[26] D. D. Roure, K. Belhajjame, P. Missier, J. M. Gómez-Pérez, R. Palma, J. E. Ruiz, K. Hettne, M. Roos, G. Klyne, and C. Goble. Towards the Preservation of Scientific Workflows. In *Proceedings of the 8th International Conference on Preservation of Digital Objects*, iPRES 2011.

[27] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics*, 5(2):51–53, June 2007.

[28] P. Trezentos, I. Lynce, and A. L. Oliveira. Apt-pbo: solving the software dependency problem using pseudo-boolean optimization. In *Proceedings of the IEEE/ACM international conference on Automated Software Engineering*, ASE '10, pages 427–436, New York, NY, USA, 2010. ACM.