An ontology supporting planning, analysis, and simulation of evolving Digital Ecosystems

Johannes Biermann
Göttingen State and University
Library, Georg-August-Universität
37070 Göttingen, Germany
biermann@sub.uni-goettingen.de

Anna Eggers
Göttingen State and University
Library, Georg-August-Universität
37070 Göttingen, Germany
eggers@sub.uni-goettingen.de

Fabio Corubolo
IPHS, University of Liverpool
Waterhouse Building, Brownlow St.,
Liverpool L69 3GL, UK
corubolo@gmail.com

Simon Waddington
Department of Digital Humanities,
King's College London
Strand, London WC2R 2LS, UK
simon.waddington@kcl.ac.uk

ABSTRACT

Digital Ecosystems (DEs) are interwoven networks of Digital Objects (DOs), policies, processes, services and user communities, within constantly changing and interacting environments. Their complex grown structures can be difficult to maintain and it can be hard to foresee the impact of planned and unplanned change.

The PERICLES EU FP7 project developed the Digital Ecosystem Model (DEM), an ontology to model those complex DEs for supporting their maintenance and preservation. It provides concepts to express dependencies, provenance and analysis. Planned or unplanned changes to a DE represented via the DEM can be simulated in advance to analyse and mitigate risks, and to assure the quality of the DE architecture.

PERICLES developed the EcoBuilder tool to support scenario experts in modelling aspects of interests of their DEs with the DEM for further investigation and maintenance. The DEM can be extended by domain specific ontologies to support various use cases of different domains, e.g. the digital media and art domains for which specialised ontologies [10] are developed in PERICLES. This approach supports documentation of the entities and their environments not only for preservation purposes, but also for the management of environmental drift in a wider range of domains, such as the digital ecosystems of Internet of Things (IoT), which we used as example in this paper.

Categories and Subject Descriptors

I.6 [SIMULATION AND MODELING (G.3)]: I.6.5 Model Development - Modeling methodologies

General Terms

Design, Theory, Management, Verification

Keywords

model driven preservation; complex environment; sheer curation; domain independent ontology; modelling; ontology; policy; digital ecosystem; modelling strategy

1. INTRODUCTION

The PERICLES project follows a Model Driven Preservation (MDP) approach [22] utilising the flexibility of ontology models to support DP activities in complex evolving DEs. The approach builds on preservation by design principles that regard Digital Preservation (DP) activities not as add-ons, but embed them into existing architectures and workflows. MDP can be applied in the context of sheer curation [12], which means that necessary activities for the DP of DOs are integrated into the workflows of the DO's creation and use environment in a lightweight, transparent way, so that users are not disturbed in their workflows. Sheer curation modelling focuses on gathering and modelling Significant Environment Information (SEI) required for later reuse and DP from the living DEs. SEI is a broader set of information about DOs, their environments, and their dependencies which are significant for a designated purpose or scenario, as described in [9]. It can be automatically populated into the models. Volatile SEI, as for example the current usage of resources at a designated time, must be collected at that precise time and is otherwise lost. Continuous SEI extraction helps to keep a model up to date, and later to analyse the impact of past changes. The DEM supports this approach by providing the components for modelling DOs, their environments, and the relations between them throughout their lifespans. It is designed to be domain agnostic and integrates well into the overall PERICLES model driven approach.

Ontology based models provide a high level of flexibility, because they are modular, reusable across different use cases, and machine-processable. This flexibility is needed to represent complex DEs which were often not planned in advance and evolved over time. The DEM does not dictate an architecture to meet the requirements of evolving DEs. Instead it provides flexible mechanisms for representing a DE including dependencies between a wider range of different entities, environments, and domains to support its comprehension, the justification of planned changes and design decisions, and to facilitate the analysis and management of the underlying DE.

DEs include entities that perform changes, such as user communities that access and reuse DOs through services, and processes that operate on DE entities. Changes can occur [4]:

- in expectations, requirements, and background knowledge of user communities
- by merging communities
- through the introduction of new policies
- through the exchange of DOs with other communities
- in processes and workflows
- in the technical infrastructure and dependencies
- in larger social or cultural contexts as laws, disciplines or cultural norms

A change can be planned or unplanned. The risks of planned changes can be mitigated through model analysis before applying the change. Introduction of change can therefore be done in a controlled way by assessing the impact on the entities in the DEM beforehand.

The DEM follows a modular approach for the DEM in which the core model is extendable by more granular sub-models, if required by a scenario. This approach reduces the overall complexity of the model and allows us to describe focus areas in more detail without overloading the model with unused entity definitions. The complexity of the models can especially be a problem at the phase of model creation. It can be controlled through the use of supporting tools and choice of an appropriate modelling strategy.

The next section introduces related work regarding the term Digital Ecosystem. Section 3 explains the DEM in more detail including its purpose, application, and structure. Strategies for modelling DEs are presented in section 4, and section 5 shows the practical application of the model based on an IoT example, and presents the EcoBuilder tool for the creation of DEMs. The paper finishes with the evaluation of the approach, outlook and future work.

2. RELATED WORK ON DIGITAL ECOSYSTEMS

The term *Digital Ecosystem* has been in use since 2000s in with different meanings. In [11] a DE consists of people that use digital technology for communication which has finally an influence on the knowledge and economics. The term has also been extended to *Digital Business Ecosystems* [18] that describe the relationship from IT systems to economical processes. There is analogy with biological ecosystems as described in [13], [18] or in the DP context [19]. The term is also used in the context of a supplier of hard- and software together with different services that connect between the components [1].

As the DEM approach is designed being domain agnostic it can be applied in interesting emerging fields such as IoT ecosystems. An IoT ecosystem can be considered as a special form of a DE which consists of network enabled devices, including sensors, smart devices, processes, computer systems, policies and user communities which can be modelled with the DEM. These business aspects of the IoT ecosystems are not currently represented in the DEM, but could be introduced using existing business ontologies, as it is done for domain specific parts in the DEM. IoT ecosystems may exist in preservation related context; either an IoT ecosystem needs to be preserved (e.g. a IoT time based media artwork) or a preservation system uses itself IoT technology for its services.

The DEM concept should be distinguished from the Enterprise Architecture (EA) and Enterprise Architecture Modelling disciplines. An EA describes the way how an enterprise works and describes business functions, such as delivery of products and services and improving their performance [21]. The aim is to achieve an optimal support from the IT systems for the business strategies and inclusion of the IT strategy into the business vision. The TIMBUS context ontology uses concepts from EA for preserving business processes [17]. Another proposal on how to use EA modelling technique for preservation systems is described in [3].

PERICLES investigates the use of DEMs to support DP use cases and builds therefore on previous work from the DP domain. Models have been used to solve various issues in the area of DP. The Preservation Network Models from the CASPAR project help to estimate the costs of preservation actions by including risks into a model which allow to compare different strategies [7]. The SCAPE project defines a preservation ecosystem consisting of policies and a controlled vocabulary [15], that is based on typical DP use cases. SCAPE's preservation policy definitions are integrated into the DEM's preservation policy extension to enable a more detailed modelling of preservation related scenarios. DEM must be distinguished from PREMIS (Preservation Metadata: Implementation Strategies)[1] because some elements of PREMIS (e.g. Digital Object, Agent) have a superficial similarity to the DEM. The main distinction is that PREMIS is a metadata standard for the description of DOs.

3. THE DIGITAL ECOSYSTEM MODEL

The DEM uses the Web Ontology Language (OWL)[2], which provides a broad support for tools and ontology reasoning. It is furthermore a specialisation of the Linked Resource Model (LRM) [23] ontology, which has been developed within the PERICLES project. The principle LRM notions are resource, agent and dependency. Resources can be used to represent DE entities for version tracking and provenance recording. The LRM enables semantics to be added to dependencies, for example to describe the impact of change through precondition and impact constructs. Due to the ontology-based nature an arbitrary level of detail can be added by importing external domain ontologies, such as the digital media and science ontologies developed at the PERICLES project [10] or CIDOC-CRM[3]. A DE is modelled through the instantiation of the abstract DEM entity templates. We call the resulting model of a DE *instantiated scenario model*. The following figure 1 shows the relation of the ontologies in a descending order of abstraction.

[1] http://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf

[2] https://www.w3.org/standards/techs/owl

[3] http://www.cidoc-crm.org

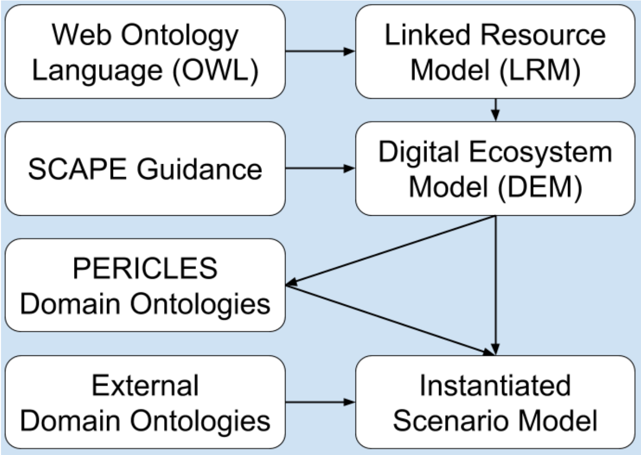

Web Ontology Language (OWL)
Linked Resource Model (LRM)
SCAPE Guidance
Digital Ecosystem Model (DEM)
PERICLES Domain Ontologies
External Domain Ontologies
Instantiated Scenario Model

Figure 1. DEM ontology relations

The DEM is partitioned into a core model and five sub-ontologies *Infrastructure*, *Analysis*, *Processes*, *Policies*, *Digital Preservation Policy* (see figure 2) which cover different aspects of the DE. The specific scenario under consideration determines which of the sub-models are used, besides the mandatory core model. The figure also illustrates the interactions of the sub-ontology with the main entities from the DEM Core model. The details of all sub-models are described in the following sections.

3.1 The DEM Core model

The core model consists of five main entity types which allow to define basic principles that are valid for all inheriting entities of the model, such as the integration of the LRM versioning mechanism for all entities, or relations which can have any DEM entity as target, e.g. "*Policy constraints*" or "*User Community owns*".

A **Technical Service** is an entity consisting of hardware and software components and their interfaces. It can be modelled in more detail by including the DEM *Infrastructure* model.

A **User Community** is a group of one or more humans that are part of the DE. This covers typically users which are interested in the use and reuse of the entities, as well as maintaining groups of people who use services for administrative tasks. It can be used similar to the OAIS concept of Designated Communities "*An identified group of potential Consumers who should be able to understand a particular set of information.[...]*" [6]. *User Communities* inherit from the *Ecosystem Agent* entity, which means that they can perform actions in the DE. *Human Agents* can be modelled as being members of *User Communities*. A *Role* entity allows to describe the role of *User Communities* or *Human Agents* in more detail.

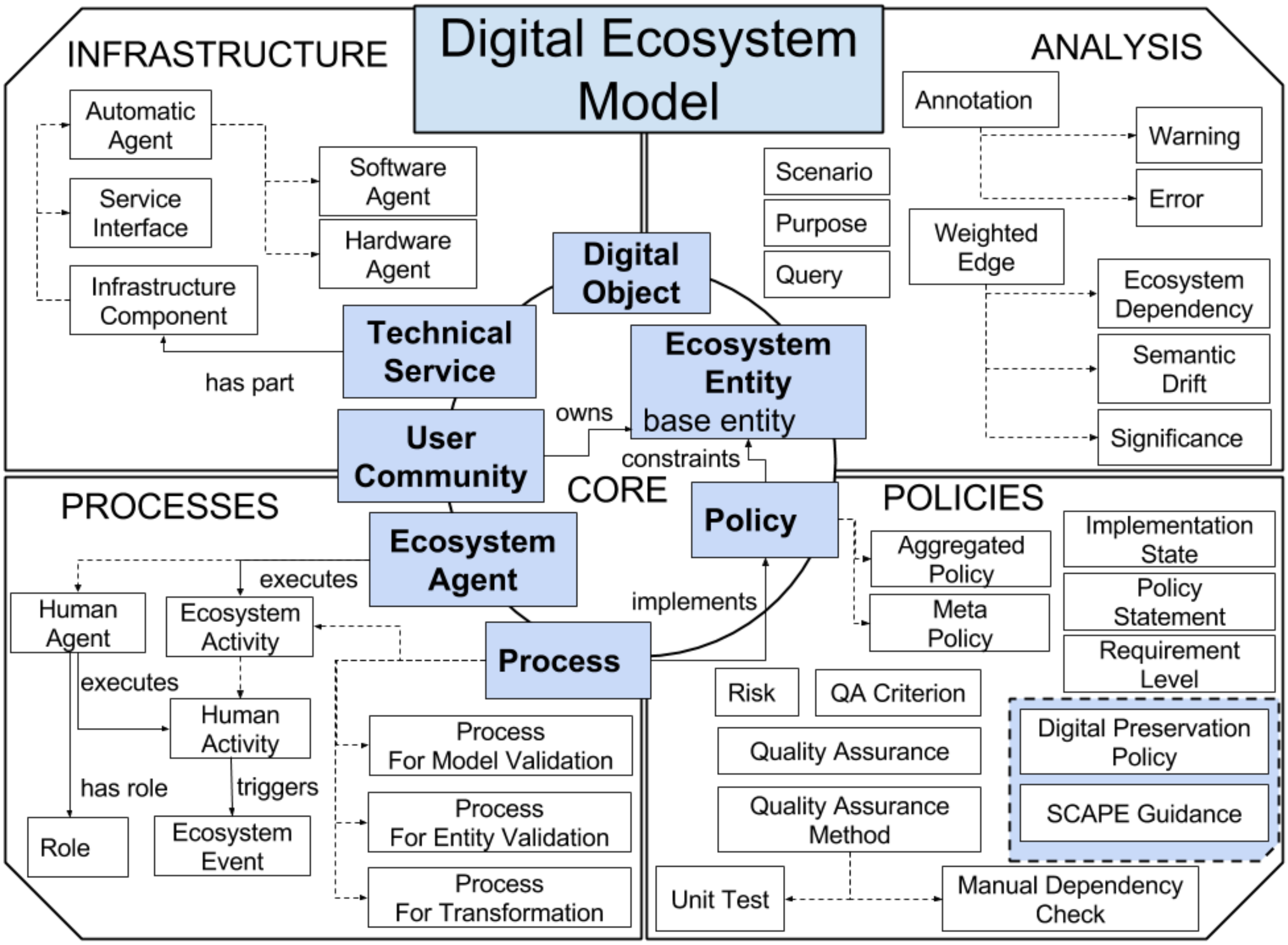


Figure 2. DEM Core model and the specialized sub-ontologies. The Policies model has an optional preservation policy extension.

The **Digital Object** entity of the DEM is intended for "*any item that is available digitally*" [24] and stands for the digital content of interest. It can be constructed as an aggregation of other *Digital Objects*, inheriting the *Aggregated Resource* definition from the LRM.

A **Policy** is a guideline or a goal that defines the desired state inside a DE, expressed with constraints. The definition we use covers mandatory, legal, aspirational and not implementable policies. Typically a policy is realised by one or more processes and a policy may constrain the behaviour and evolution of any entity in the DEM.

A **Process** transforms an input to a certain output by linking activities and can involve automatic procedures and human interactions. It can be regarded as operationalisation of policies. Process internals are usually not modelled inside the DEM, instead a link to suited implementations can be added to the entity (e.g. Business Process Model and Notation (BPMN)).

3.2 The DEM Policy model and its extension for preservation policies

The DEM includes an implementation of a theoretical policy model which was developed during the PERICLES project and described in [20]. The DEM Policy model enables the creation of a policy-driven DEM instance, in which policies are central drivers of the infrastructure. They can be used to describe legal and institutional requirements, including aspirational policies representing the direction and aims of the organisation. Policies can be expressed in formal or non-formal languages, and can define methods for quality assurance to validate their correct implementation in the DE.

Policy entities can be modelled as aggregations of other *Policies* inheriting from the LRM *Aggregated Resource*, so that aggregation trees can be defined using policy derivation guidelines [20], [2]. This allows to break complex policies into simpler ones which are easier to handle for analysis and reuse.

Policies can be linked to their implementing processes via the "*is enforced by*" relation. The "*constraints*" relation allows to model which entities are affected by a policy, and the "*target community*" relation can link to a target audience modelled as *User Community*. *User Communities* can also be modelled as being mandators of a policy, and a *Human Agent* can be defined as its "*responsible person"*.

Details of a policy are modelled using the *Implementation State, Policy Type,* and *Requirement Level* entities or by using attributes to assign the current validity state, or to identify conflicts or risks. If policies are modelled in detail, which means that they contain executable queries or links to executable processes, the model can act as a tool for policy verification.

The DEM Policy model can be extended by a more specialised sub-ontology for expressing preservation policies which make use of the SCAPE policy framework[4], including the classification of SCAPE Guidance Policy[5].

3.3 The DEM Process model

The main purpose of the DEM Process model is to model which *Ecosystem Entities* are processed in which way, whereby the results of this processing are defined through the "*has input / has output"* relations, and the executing *Ecosystem Agent* is linked to the *Process* entity via the *"executes"* relation. The bidirectional relation "*is implementation of"* relates a *Process* entity to a corresponding *Policy*. Processes can be regarded pure technically and related through the "*runs on"* relation to a *Technical Service,* or involve the interaction of *Human Agents.*

The DEM Process Model includes relationships of agents, events and activities. An agent is anything (human or machine) that introduces change into the DE. Agents raise events and activities act as an effect of events. The concept of activities having states (started, stopped, paused, resumed) and bindings to time intervals are inherited from the LRM. A method on how to use the DEM to automatically combine linked sub-processes into a complete preservation process is described in [5].

Similar to *Policies*, the *Process* entity inherits from the LRM *Aggregated Resource,* and can be an aggregation of sub-processes, which have a sequence number related to their execution position. This allows to model designated inputs and outputs of specific sub-processes to depict a high level data flow.

The model contains dedicated process types for adding automated validation of the model regarding custom defined requirements. The *Process for Model Validation* selects a subset of entities that needs to be validated, and passes each of these entities to a *Process for Entity Validation*. A *Process for Entity Validation* validates a single entity against the requirements. If the entity is not valid, it is passed to a *Process for Transformation*. This process executes the required steps to bring the entity into a valid state. These steps could involve the interactions of a person, who needs to be notified by the process.

3.4 The DEM Infrastructure model

The DEM Infrastructure model covers entities of the technical infrastructure, in particular to model *Technical Services* and their relations in detail. *Technical Services* are LRM *Aggregated Resources* that consists of *Infrastructure Components.* Their software and hardware components can be expressed with the *Software Agent* and *Hardware Agent* entities*.* The *Service Interface* is a specific *Infrastructure Component,* that is used to model the way how entities, such as DOs and technical services, can be accessed, and who accesses them. This allows for example expressing access rights.

The DEM Infrastructure model supports the management of technical components for a better planning of infrastructure changes. It can be an aid to identify single points of failure in the infrastructure architecture, necessary updates, policy violations, or scarcity of resources. Simulations of failures, replacements, or updates of the components enable a higher quality of change planning. The model can serve as documentation for the infrastructure, which facilitates the creation of similar DEs, or the emulation of environments of DOs.

3.5 The DEM Analysis model

The DEM can be regarded as a multigraph in which the entities are nodes and the relations are edges. The DEM Analysis model is specifically designed to add information layers on top of a DEM instance. This information allows to analyse the model with arbitrary graph analysis algorithms.

[4] http://wiki.opf-labs.org/display/SP/SCAPE+Policy+Framework

[5] http://wiki.opf-labs.org/display/SP/Policy+Elements

Weighted Edges (WE) can be modelled between entities to introduce weights and annotations, via the *Annotation* entity. A WE can be used to introduces weighted graph entities by letting it point from and to the same *Ecosystem Entity* to add arbitrary weights and annotations to that entity. WEs are directed, equivalent to the LRM "*from*" and "*to*" dependencies and build on the concept of the Weighted Dependency Graphs [8].

The *Scenario* entity enables a view on the ontology considering a subset of entities, which allows the usage of one DEM instance for different use cases, e.g. to highlight certain procedures or considering only a subset of entities. The *Purpose* entity describes the intention of entities or scenarios.

The *Semantic Drift* entity expresses drifts between different versions of entities or between entities with semantic similarities. This kind of analysis is called ontology evolution [16], [14] which compares instances of the ontology. Each entity can be weighted with a *Significance* value for a specific purpose or scenario [9], [8]. It is also possible to use the weight attribute of a dependency to model it as being optional, or annotate dependencies with rules which constrain their appliance.

4. MODELLING STRATEGIES

A modelling strategy is a plan that defines when designated entity types and relations can be added and the order and level of information detail introduced into the model. The choice of a modelling strategy depends on the use case and defines the required level of detail and the order of modelling. The creation of detailed models can be costly and complex, and continuous updating is required to keep the model up to date. There is a trade-off between the introduction of more detail, and the required effort for its modelling and maintaining. A good strategy is to start modelling with the most significant entities and relations for a scenario, and to work one's way forward to less significant details only when needed.

A high level of detail has an impact on the effort needed to create, analyse and maintain the model. It is never sure which information will be needed for future reuse and analysis, therefore the amount of unnecessarily stored information will also increase with the level of detail. On the other hand, a more detailed modelling can support future analysis and information reuse, which is not foreseen at the moment of modelling and which could open up new possibilities.

In this section we describe different approaches to DE modelling, that can be used separately or together depending on the use case and scope of the DE, and give an indication of where the specific type of modelling could apply. The DEM can be used in all of the three presented strategies, as it provides the components for modelling, but it is not enforcing a modelling strategy.

4.1 Top-down and bottom-up approach of modelling

A top-down approach starts by modelling a generic view covering the most relevant entities and relations of a scenario. On this generic basis the model is refined until the level of detail reaches the necessary value for representing a given scenario. This approach produces an abstract DEM, which can provide views on many different aspects of the overall architecture. A specific sub-type of this modelling strategy is the policy based modelling described later in this section.

The bottom-up approach works just opposed starting from very detailed information about single DE components, which are later connected to abstract entity constructs. This approach has an initial view on computer system environments in which the DOs are created and changed especially on components, software and other technological aspects. Raw information which is significant for the user's purpose can be gathered from such an environment, e.g. by the PERICLES Extraction Tool [9], but it needs to be described by assigning it to the more abstract ontology constructs to make it useable.

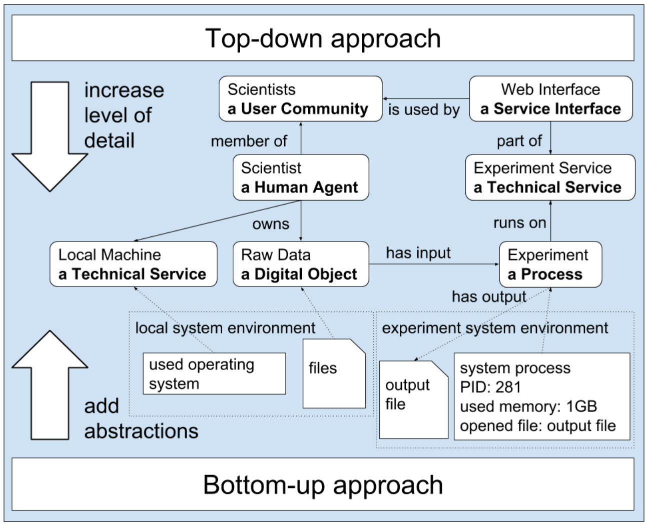


Figure 3. Convergence of top-down and bottom-up approach

The two approaches can be combined to create an overall generic view of the DE enriched with detailed information about the entities and their local environments as illustrated in figure 3. Bottom-up is the view from the local system of the scientist on the system files and environment information. More bottom-up models can be created from the information of other environments in parallel. At the beginning there is no connection between the models describing different environments. The connection which combines the environment models to a whole DEM has to be introduced by adding abstractions and relations to the models.

A top-down approach would model the abstract entities like the scientist, communities and services, first. Starting from these abstractions the level of detail is enriched with information until the model becomes meaningful for analysis.

It is possible to start both approaches in parallel at the same time, and to create the connection between the different models by letting them grow together. A combined model can be used for complex analysis of detailed information through the overall DE. The model shown in figure 3 for example would allow to analyse the data flow of real existing files though the DE.

4.2 Policy based modelling

Another approach that can be classified as top-down, is starting the modelling from an DE's policies (see section 3.2), identify other DE entities and draw the relations between them. Since most of the DE entities will be constrained by some broadly defined policy, this can offer an effective guidance in DE modelling from the intentions and objectives to their concrete implementation. A DEM can illustrate how policies map to concrete infrastructure and requirements, as well as provide effective means of validating their implementation on deployment and change.

5. USE OF THE DEM AND EXAMPLE

PERICLES developed a tool that provides a simple way for the creation of DEMs, the EcoBuilder[6]. It enables users, who are non-ontology experts, to model their DEs and scenarios. The Java tool is Open Source under Apache v2 license, and uses the Apache Jena[7] library for the creation of ontologies.

The EcoBuilder provides templates for the well-defined creation of DEM entities and relations that can be accessed via two interfaces: A Java API for developers, and a graphical user interface for use case specialists. It generates the ontology resources and relations in Turtle[8] and RDF/XML[9] format from the user's input. The user is not bound by the EcoBuilder to follow a designated modelling strategy, but we recommend a structured modelling strategy as described in the modelling strategies section. This helps to keep an overview of complex scenarios and entity linkage.

In contrast to well-known generic ontology modelling tools such as Protégé[10], the EcoBuilder reduces the modelling complexity to a level on which the tool can easily be used by scenario experts without the necessity of in-depth ontology knowledge. The EcoBuilder ensures that well defined modelling techniques are used to create a mature and clear scenario model. However, the complexity reduction restricts the modelling, as not all possible modelling techniques and available resources can be used with the templates. They can be added to the generated models afterwards, if required for the scenario. This proceeding is recommended especially if further ontologies should be imported for domain specific refinements.

The following figure 4 shows an IoT example for which a DEM will be created using the EcoBuilder.

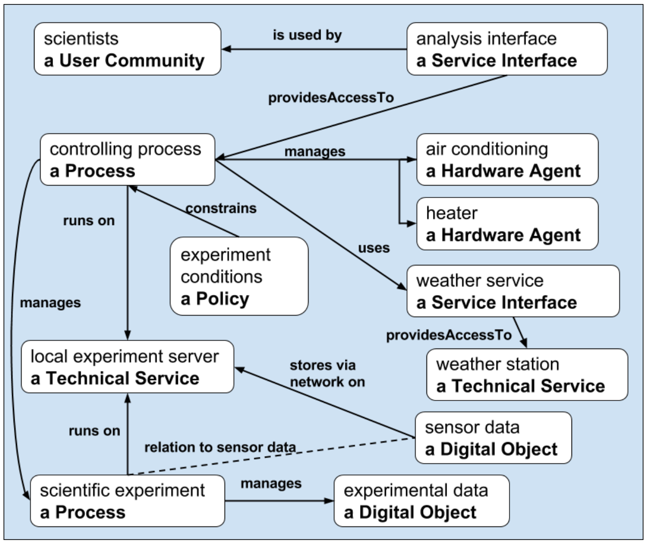
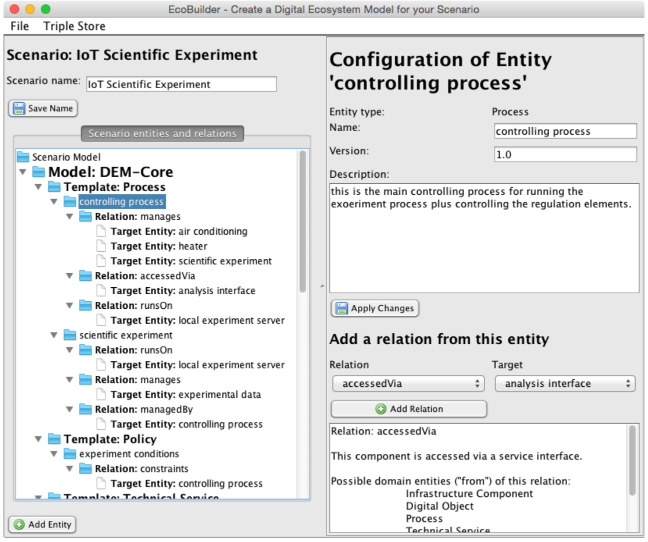


Figure 4. The first iteration for a DEM of the scientific experiment

A scientific experiment is sensitive to temperature and humidity changes. Sensors are placed inside the experimental object and the room. There is a controlling process which is responsible for the overall experiment execution. It sends control commands to an air conditioner and heater for regulation. The scientific experiment is a process that executes the experiment itself and collects the data. All devices are connected via a network. The controlling process also takes into account the data from the local weather station for better control planning, because the regulation elements do not react instantly. If the parameters are out of range, the experiment is paused. This description allows to identify components and relation in a top-down way for a first version of an instantiated DEM, as figure 4 shows.

Depending on the desired level of granularity, more details can be added, if needed. For example the experimental data could be modelled as aggregated resource consisting of humidity and temperature data, or a domain specific weather ontology could be imported into the generated DEM. EcoBuilder can be used to create the DEM model instance from figure 4. A screenshot of the tool is shown in figure 5. On the left side there is a tree which depicts the entities of the DEM instance. On the right half details of an entity can be entered and relation between entities can be created.


EcoBuilder - Create a Digital Ecosystem Model for your Scenario
File
Triple Store
Save Name
Scenario entities and relations
Apply Changes
Add Relation
Add Entity

Scenario: IoT Scientific Experiment

Scenario name: IoT Scientific Experiment

Scenario Model
- Model: DEM-Core
  - Template: Process
    - controlling process
      - **Relation:** manages
        - **Target Entity:** air conditioning
        - **Target Entity:** heater
        - **Target Entity:** scientific experiment
      - **Relation:** accessedVia
        - **Target Entity:** analysis interface
      - **Relation:** runsOn
        - **Target Entity:** local experiment server
    - scientific experiment
      - **Relation:** runsOn
        - **Target Entity:** local experiment server
      - **Relation:** manages
        - **Target Entity:** experimental data
      - **Relation:** managedBy
        - **Target Entity:** controlling process
  - Template: Policy
    - experiment conditions
      - **Relation:** constraints
        - **Target Entity:** controlling process
  - Template: Technical Service

Configuration of Entity 'controlling process'

Entity type: Process
Name: controlling process
Version: 1.0
Description:
this is the main controlling process for running the exoeriment process plus controlling the regulation elements.

Add a relation from this entity

Relation: accessedVia
Target: analysis interface

Relation: accessedVia

This component is accessed via a service interface.

Possible domain entities ("from") of this relation:
Infrastructure Component
Digital Object
Process
Technical Service

Figure 5. Screenshot from EcoBuilder modelling the scientific experiment

The resulting DEM can be used for static model analysis, for example weights and annotations for arbitrary purposes can be added to the components or possible problems can be identified. In the described example the controlling process seems to be more important than the weather service, because of the great amount of other entities depending on it. From the structure and relations of the entities we can identify that the weather data is not recorded and there is no policy in place in case of an unavailable weather web service. This induces the following planned change:

A new policy for handling a dysfunctional weather service is introduced. It says that after 5 minutes of non-reachability the experiment is started without weather information. From figure 4 we can see that the controlling process is the only component which accesses the weather web service. Therefore it is the only component that has to be changed and the change will not have any side effects to the remaining DE.

[6] https://github.com/pericles-project/EcoBuilder

[7] https://jena.apache.org

[8] https://www.w3.org/TR/turtle/

[9] https://www.w3.org/TR/owl-xmlsyntax/

[10] http://protege.stanford.edu/

Unplanned change occurs for example if the heater entity cannot be read anymore. From the DEM we can identify that the dependency to the controlling process is broken which affects also the dependencies to the scientific experiment, therefore the experiment cannot run anymore.

Besides functional details of the IoT example, the model can be used to analyse and introduce DP related functionality. In particular the availability of the scientific data and the scientific process are candidates for adding DP related functionality. This is a planned change which involves that processes, policies and technical systems for these two components are added to the model.

6. EVALUATION

The literature survey has shown that there are other models for expressing DP related issues, as listed in the related works section. Some cover very specific issues, like providing a cost based model, defining a policy or metadata model. More general models build up on the EA approach for modelling preservation systems. EA modelling is limited in expressing semantic of change, a change is done by providing "as is" "to be" models. The concepts of dependency are therefore motivated by the need to express realisation of business functions and services, rather than providing the semantics required to reason on change. In contrast the DEM is follows a more general approach, but does not provide designated support for expressing business and cost based flows.

The DEM is domain agnostic, enables the modelling of complex DEs and provides support for change and other graph-based analysis. Such analysis can reveal for example single points of failure, or simulate the impact of a change. Some evaluations require a regularly updated model to enable history tracking of evolving DEs and to deduce past and present modifications or to indicate trends. There are different techniques for measuring differences of ontologies [14]. At a wider scope the comparison of different DEM instances enables the analysis of consistency, i.e. infer the compatibility and similarity of policies or procedures of object management or the users involved. Well-proven instances could become a collection of best practises and they could act as a communication tool.

However the integration into processes that help to provide model updates is only a proof of concept for the demonstration of the PERICLES use cases. There have been experiments on mapping the output of PERICLES Extraction Tool[11], a tool for information gathering, into a DEM scenario. It can enrich information about instances of the entities or deliver new entities. In general more work is required on the analysis part of the modelled DEM ontologies and this needs to be verified with real world examples. This requires a greater automation of the ontology population, as mentioned before.

The EcoBuilder tool enables people who are not familiar with writing ontologies or source code to create DEMs, however it is limited in its current state. The graphical user interface is designed to fit the needs of the PERICLES ontologies, and it is currently not possible to import external ontologies using the tool. Therefore other (e.g. domain specific) ontologies have to be integrated into the created DEM by hand. Plans for a further development of the tool involve addressing this issue.

On the one hand a graphical user interface for modelling simplifies the creation of well-defined models through templates, on the other hand it intentionally limits the possible level of modelling. By the creation of models for designated scenarios we had the best results using the EcoBuilders Java API, because it simplifies the modelling as one has not to write the ontology triples itself, and leaves at the same time enough freedom for customised adaptations. Unfortunately this modelling approach is not suitable for non-programmers.

7. CONCLUSION

This paper presented the DEM ontology, which follows a domain independent approach for modelling DEs, taking into account dependencies and different types of entities and their interactions. The presented EcoBuilder tool supports straightforward modelling of DEMs. Models created for a specific scenario can support preservation purposes, as well as the maintenance and analysis of planned and unplanned change of complex DEs, which is otherwise hardly to perform in a controlled way. The presented model can further be utilised to analyse semantic drift, ontology evolution, single point of failures, dependency validation and architecture changes. For this purpose, the DEM can be kept synchronised with the underlying ecosystem through monitoring and extraction of information about ecosystem changes.

The DEM is divided into six abstract OWL based ontologies. A mandatory core model contains a set of base entities. It can be extended by the DEM Policy model, the SCAPE preservation policies, the DEM Process model, the DEM Infrastructure model, and the DEM Analysis model; depending on the scenario requirements in order to model different aspects of a DE.

This paper further investigated modelling strategies to simplify the modelling process of DEs. A top-down strategy starts the DE entity creation from a generic model which is refined to the scenario required level of detail. In contrast a bottom-up approach models the detailed entities first and connects them via abstract relations afterwards. Top-down modelling applied with the EcoBuilder was illustrated based on an example that analyses single point of failures and architecture changes of an IoT ecosystem for quality assurance.

The presented model driven approach facilitates the maintenance and preservation of DEs, because it provides a well-defined way to deal with their interwoven structures. Its application is enabled by the supporting tools and modelling strategies.

8. ACKNOWLEDGMENTS

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 601138.

9. REFERENCES

[1] Bajrin, B. Why It's All About the Digital Ecosystem. 2011. https://techpinions.com/why-its-all-about-the-ecosystem/4567.

[2] Bechhofer, S., Sierman, B., Jones, C., Elstrøm, G., Kulovits, H., and Becker, C. Final version of Policy specification model. 2013. http://scape-project.eu/wp-content/uploads/2013/08/SCAPE_D13.1_UNIMAN_V1.0.pdf.

[11] https://github.com/pericles-project/pet

[3] Becker, C., Antunes, G., Barateiro, J., Vieira, R., and Borbinha, J. Modeling digital preservation capabilities in enterprise architecture. ACM Press (2011), 84.

[4] Biermann, J., Ludwig, J., Carter, A., et al. Deliverable 5.1.1 Initial Report on Preservation Ecosystem Management. 2014. http://pericles-project.eu/deliverables/12.

[5] Campos-López, N. and Wannenwetsch, O. The PERICLES Process Compiler: Linking BPMN Process into Complex Workflows for Model-Driven Preservation in Evolving Ecosystems. 2016.

[6] CCSDS Secretariat, ed. *Reference Model for an Open Archival Information System (OAIS)*. 2012.

[7] Conway, E., Matthews, B., Giaretta, D., Lambert, S., Wilson, M., and Draper, N. Managing Risks in the Preservation of Research Data with Preservation Networks. *International Journal of Digital Curation 7*, 1 (2012), 3–15.

[8] Corubolo, F., Eggers, A., Hasan, A., et al. Deliverable 4.1 Initial version of environment information extraction tools. 2014. http://pericles-project.eu/deliverables/17.

[9] Corubolo, F., Eggers, A., Hasan, A., Mark Hedges, and Waddington, S. A pragmatic approach to significant environment information collection to support object reuse. *Proceedings of the 11th International Conference on Digital Preservation*, (2014), 249–258.

[10] De Weerd, D., Mitzias, P., Riga, M., Muller, C., and Haage, K. Deliverable 2.3.2 Data Surveys and Domain Ontologies. 2015. http://pericles-project.eu/deliverables/48.

[11] Fiorina, C. THE DIGITAL ECOSYSTEM. 2000. http://www.hp.com/hpinfo/execteam/speeches/fiorina/ceo_worldres_00.html.

[12] Hedges, M. and Blanke, T. Digital Libraries for Experimental Data: Capturing Process through Sheer Curation. In T. Aalberg, C. Papatheodorou, M. Dobreva, G. Tsakonas and C.J. Farrugia, eds., *Research and Advanced Technology for Digital Libraries*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, 108–119.

[13] Khalil, M.A.T., Dominic, P.D.D., Bin Hassan, M.F., and Mushtaq, A. An In Depth Analysis of Ecosystems & Blueprint of Digital Business Ecosystem (DBE) Framework for Malaysian SMEs. *Computer Science Journal Volume 1*, Issue 1 (2011).

[14] Kontopoulos, E., Riga, M., Mitzias, P., et al. Deliverable 4.4 Modelling contextualised semantics. 2016. http://pericles-project.eu/deliverables/76.

[15] Kulovits, H., Kraxner, M., Plangg, M., Becker, C., and Bechhofer, S. Open Preservation Data: Controlled vocabularies and ontologies for preservation ecosystems. *Proceedings of the 10th International Conference on Preservation of Digital Objects (iPRES 2013) 3-5 September 2013, Lisbon, Portugal*, (2013).

[16] Maronidis, A., Chatzilari, E., Kontopoulos, E., et al. Deliverable 4.3 Content Semantics and Use Context Analysis Techniques. 2016. http://pericles-project.eu/deliverables/63.

[17] Mayer, R., Antunes, G., Caetano, A., Bakhshandeh, M., Rauber, A., and Borbinha, J. Using ontologies to capture the semantics of a (business) process for digital preservation. *International Journal on Digital Libraries 15*, 2–4 (2015), 129–152.

[18] Nachira, F., Nicolai, A., Dini, P., Le Louarn, M., and Rivera Leon, L. *Digital business ecosystems*. EUR-OP, Luxembourg, 2007.

[19] Pocklington, M.J., Eggers, A., Corubolo, F., Ludwig, J., Hedges, M., and Darányi, S. A Biological Perspective on Digital Preservation. *Proceedings of the 11th International Conference on Digital Preservation*, (2014).

[20] Simon Waddington, Tonkin, E., Gill, A., et al. Deliverable 5.2 Basic tools for Digital Ecosystem management. 2015. http://pericles-project.eu/uploads/files/PERICLES_WP5_D5_2_Basic_Tools_for_Ecosystem_Management_V1_0.pdf.

[21] Versteeg, G. and Bouwman, H. Business architecture: A new paradigm to relate business strategy to ICT. *Information Systems Frontiers 8*, 2 (2006), 91–102.

[22] Vion-Dury, J.-Y. PERICLES: towards model-driven preservation. 2015. http://pericles-project.eu/publications/44.

[23] Vion-Dury, J.-Y., Lagos, N., Kontopoulos, E., et al. Designing for Inconsistency – The Dependency-Based PERICLES Approach. In T. Morzy, P. Valduriez and L. Bellatreche, eds., *New Trends in Databases and Information Systems*. Springer International Publishing, Cham, 2015, 458–467.

[24] Linked Heritage Glossary. https://elearning.unipd.it/cab/mod/glossary/view.php?id=5&mode=cat&hook=3.