Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval

Preservation Information Conceptual Model (and Knowledge Management)

Presentation by: Yannis Marketakis

Yannis Tzitzikas, Vassilis Christophides, Grigoris Antoniou, Dimitris Kotzinos, Giorgos Flouris, Yannis Marketakis, Stamatis Karvounarakis, Dimitris Andreou, Yannis Theoharis

FORTH-ICS

CASPAR

Outline

- Motivation & Introduction
- Aspects of Preservation
- Preservation of Intelligibility (and OAIS)
  - Formalizing the Problem
  - Intelligibility Gap
- Modeling and Preserving Provenance
  - CIDOC CRM
- Knowledge Manager (one of the CASPAR key components)
  - Responsibilities
  - SWKM
  - GapManager
- Available Data and Current Deployments
- GapManager and Data Holders

Important Questions

- Important questions
  - *What digital information preservation is?*
  - *What kind (and how much) representation information do we need? How this depends on the Designated community?*
  - *What kind of automation could we offer?*
- CASPAR contribution
  - A formalization of **intelligibility** and intelligibility gap through the notion of dependency. This approach can provide some answers to the above questions.

The problem of Preserving the Intelligibility and Provenance

Phaistos disk (dated to 1700 BC)

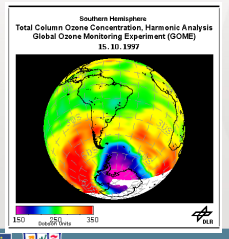
We still cannot understand it
(the meaning has not been preserved)

We still don't know how the pyramids were constructed.
(the process has not been preserved)

The problem of Preserving the Intelligibility and Provenance

How can we be sure that in the future one would be able to understand this byte stream?

1001101100001101110110111011100101111001111



- ***How this image has been derived?***
- ***When and by whom it was taken?***
- ***How the satellite image was processed (by what algorithms and with what parameters)?***

Aspects of Preservation

But what should we preserve?

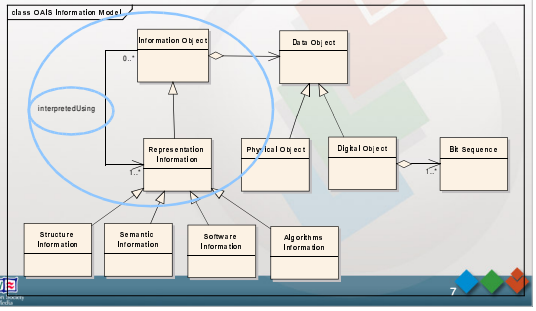
- For sure we have to preserve the **bits** of the digital objects

We should also try to preserve the **information** carried by the digital objects

- Their accessibility
- Their integrity
- Their authenticity
- Their provenance
- Their intelligibility (by human or artificial actors)

OAIS and Intelligibility

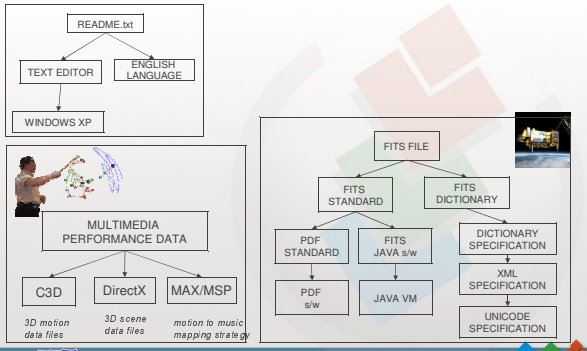
- According to OAIS, metadata are distinguished to various categories.
- One very important is that of ***Representation Information***
  - **Aim at enabling the conversion of a collection of bits to something useful**



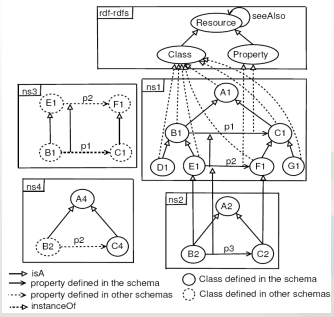
Modules and Dependencies

- In order to abstract from the various domain-specific and time-varying details, we introduce the general notions of **Module** and **Dependency**.
- **Module**
  - **We adopt a very general definition. A module could be:**
    - a piece of software/ hardware module.
    - a knowledge model expressed explicitly and formally (e.g. an Ontology)
    - a knowledge model not expressed explicitly (e.g. GreekLanguage)
  - **(the only constraint is that modules need to have a unique identity)**
- **Dependency**
  - **A module t depends on t', written t>t', if t requires t'**
  - **The meaning of a dependency t > t'**
    - t cannot function/be understood/managed without the existence of t'
    - in general we can support several dependency types (they are goal-directed)
- Note:
  - **We model the RI requirements of OAIS as *dependencies* between modules.**

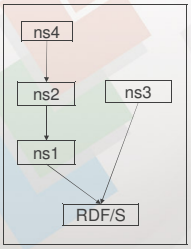
Modules and Dependencies: Examples



Modules and Dependencies: Examples (Semantic Web data)

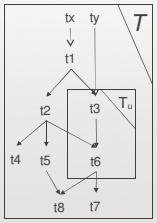


modules and dependencies



Formalizing Actor/Community knowledge (in terms of modules and dependencies)

- Each actor or community u can be characterized by a ***profile*** Tu that contains those modules that are assumed to be available/known to u.
- Formalization: $Tu \subseteq T$ (where T is the set of all modules)

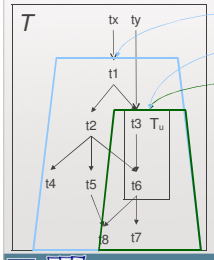


Examples

- u is an artificial agent
  - **Tu may include the software/hardware modules available to it**
- u is a human,
  - **Tu may include modules that correspond to implicit knowledge**

The notion of closure (of modules and profiles)

- Closure of a module t: C(t) = all modules on which it depends
- Closure of a set of modules S: $C(S) = \cup \{ C(t) \mid t \in S \}$
- Required modules of t: $C^+(t) = C(t) - \{t\}$



C (tx) = C(tx)- {tx}

C (ty) = C(ty)-{ty}

Closure of Tu

Let user **u** be a user with profile **Tu**

Let **t** be a module

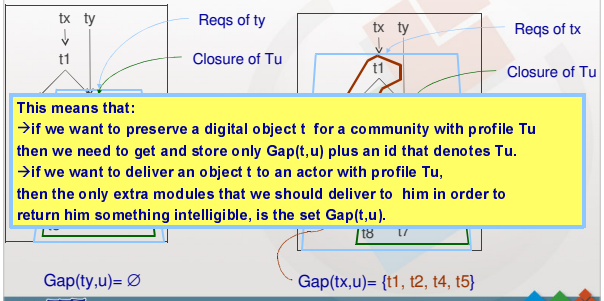
Intelligibility Gap:

The smallest set of extra modules that u needs to have in order to understand a module t.

Notation: **Gap(t,u)**

Intelligibility and Intelligibility Gap (I)

- u can understand t iff: $C^+(t) \subseteq C(Tu)$
- The intelligibility gap: $Gap(t,u) = C^+(t)-C(Tu)$



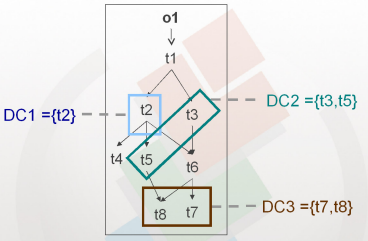
This means that:
→if we want to preserve a digital object t for a community with profile Tu then we need to get and store only Gap(t,u) plus an id that denotes Tu.
→if we want to deliver an object t to an actor with profile Tu, then the only extra modules that we should deliver to him in order to return him something intelligible, is the set Gap(t,u).

Gap(ty,u)= ∅

Gap(tx,u)= {t1, t2, t4, t5}

Exploiting DC Profiles for constructing the "right" AIPs (intelligible and redundancy free)

DC profiles could be exploited so that to be able to derive different AIPs and DIPs for different DCommunities (If the dependencies are available this could be done automatically)

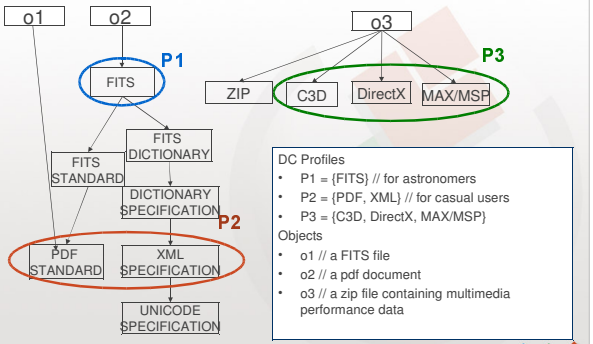


AIP of o1 wrt DC1
Object = o1
DCprofile = DC1
deps = {t1,t3}

AIP of o1 wrt DC2
Object = o1
DCprofile = DC2
deps = {t1,t2,t4}

AIP of o1 wrt DC3
Object = o1
DCprofile = DC3
deps = {t1,t2,t3,t4,t5,t6}

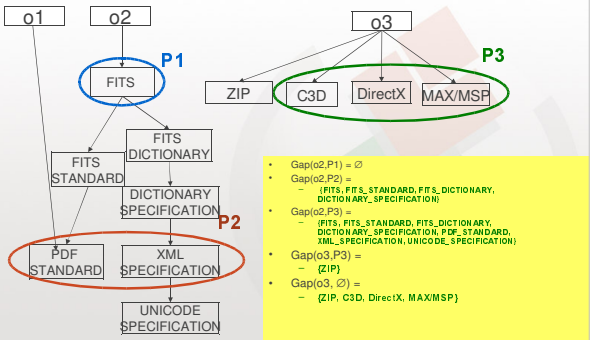
Scenario: Intelligibility-aware Packaging



DC Profiles
- P1 = {FITS} // for astronomers
- P2 = {PDF, XML} // for casual users
- P3 = {C3D, DirectX, MAX/MSP}

Objects
- o1 // a FITS file
- o2 // a pdf document
- o3 // a zip file containing multimedia performance data

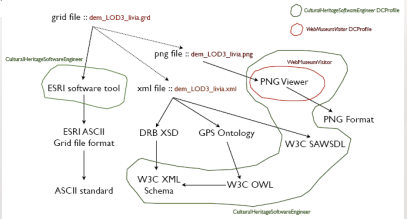
Scenario: Intelligibility-aware Packaging



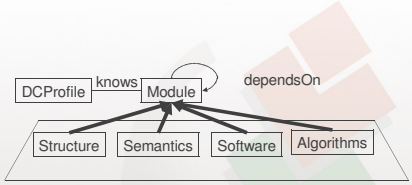
- Gap(o2,P1) = ∅
- Gap(o2,P2) =
  - {FITS, FITS_STANDARD, FITS_DICTIONARY, DICTIONARY_SPECIFICATION}
- Gap(o2,P3) =
  - {FITS, FITS_STANDARD, FITS_DICTIONARY, DICTIONARY_SPECIFICATION, PDF_STANDARD, XML_SPECIFICATION, UNICODE_SPECIFICATION}
- Gap(o3,P3) =
  - {ZIP}
- Gap(o3, ∅) =
  - {ZIP, C3D, DirectX, MAX/MSP}

Example from cultural testbed

ASCII grid file



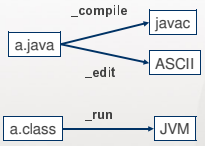
Modeling Modules and Dependencies (usage of SW languages for extensibility / inheritance)

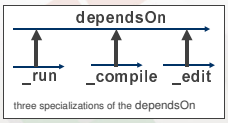


Extensible typology of modules and dependency types

Example: Extending the typology of dependencies

The **goal** determines the dependencies. For this reason we allow specializing the dependency relation





Assume a DC profile pr1 = {ASCII}

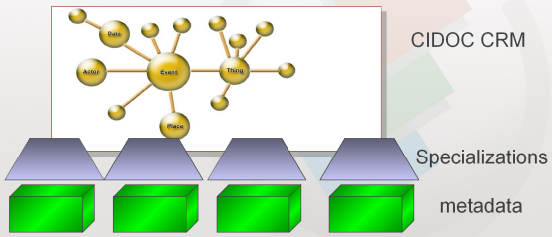
```
gap(a.java, pr1) = {javac}     // not type of dependency is specified (so all count)
gap(a.java, pr1, _edit) = ∅    // here the type of dependency is specified
```


Aspects of Preservation

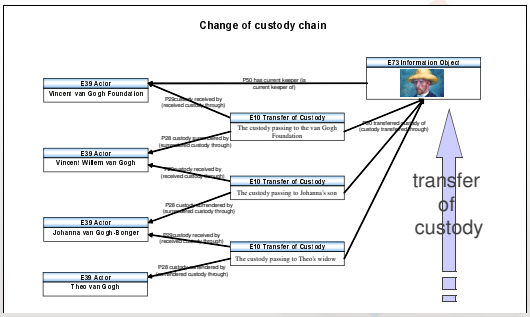
- For sure we have to preserve the **bits** of the digital objects
- We should also try to preserve the **information** carried by the digital objects
  - Their accessibility
  - Their integrity
  - Their authenticity
  - Their **provenance**
  - Their intelligibility (by human or artificial actors)

Modeling Descriptive Metadata (Provenance, Context, etc)

- Contributions in extending and specializing CIDOC CRM
  - **CIDOC Conceptual Reference Model ISO/FDIS 21127**
- Can be used as the conceptual backbone for descriptive metadata



Example of modeling provenance



Example of modeling provenance

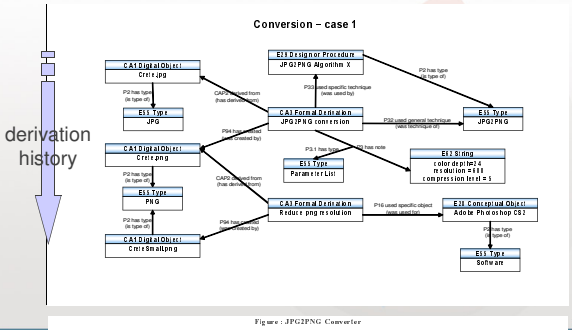


Figure : JPG2PNG Converter

Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval

(Knowledge Manager)
Component Description

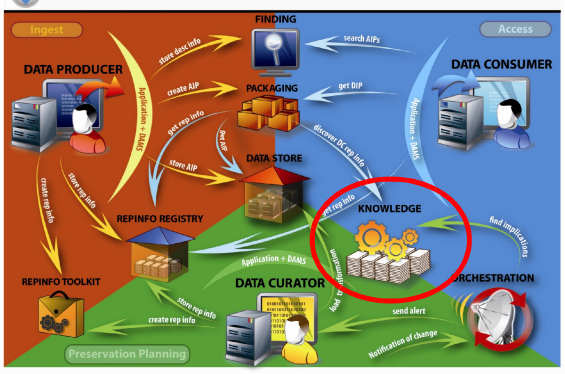
Component responsibility

Responsibilities

- Capture Higher level Semantics
- Manage Designated Community Knowledge profile
- Identify RepInfo Gaps
- Manage Ontologies and Metadata

It comprises two layers

- A) **SWKM** (Semantic Web Knowledge Middleware) which is the lower layer providing a set of core services for managing Semantic Web data.
- B) **GapManager** which is the upper layer providing high level services based on the OAIS model aiming at offering an abstraction useful for preservation information systems.



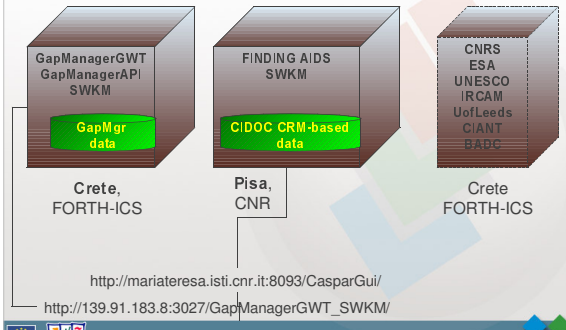
Available Data and Current Deployments

Available data

ontologies **and** data

- Intelligibility-related
  - **Core Ontology** for the Gap Manager
  - Instantiations of the ontology
    - exported from the PRONOM registry
    - from various testbeds
    - from the Registry
- Descriptive metadata
  - **CIDOC CRM ontology**
  - **Specializations of CIDOC CRM (e.g. FRBRoo, testbed-related)**
  - **Instantiations of the above**
    - contemporary arts testbed (Distance Liquide, Avis ...), cultural testbed

Available Data and Current Deployments



GapManager

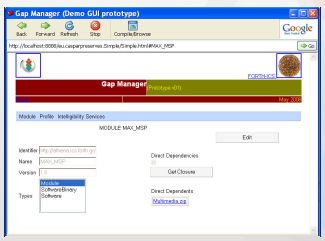
In brief, Gap Manager can aid the following tasks

- the decision of what metadata need to be captured and stored
- the identification of the data objects that are in danger in case a module (e.g. a software component or a format) is becoming obsolete (or has been vanished),
- the reduction of the metadata that have to be archived, or delivered to the users (as response to queries) on the basis of the designated community.

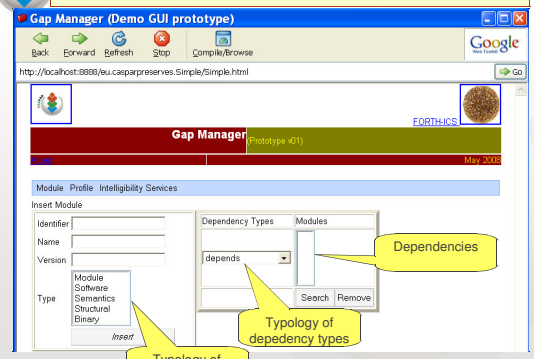
(for more details see D2102, the slides
http://wiki.casparpreserves.eu/pub/Main/PreparationForMonth24EuRevi
and the video tutorial)

GWT-based prototype

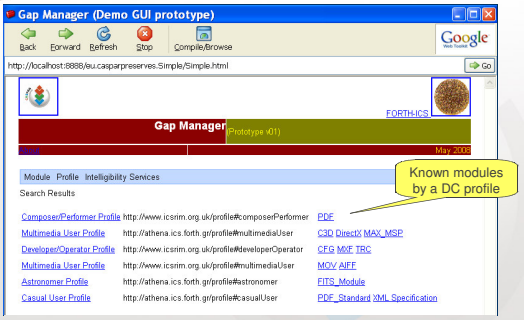
- GapManager is implemented as software component offering a plethora of methods for managing dependencies and profiles
- The implementation of an indicate prototype application using GWT is ongoing



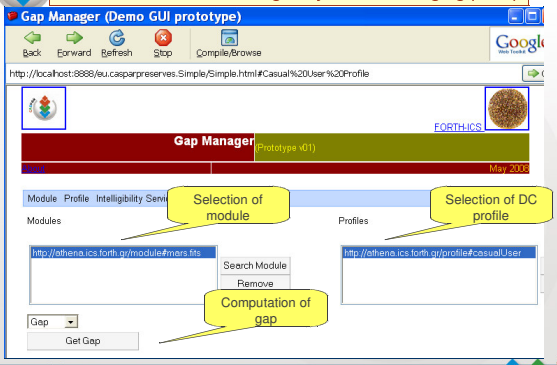
GWT-based prototype
Ingestion of Modules and specification of their Dependencies



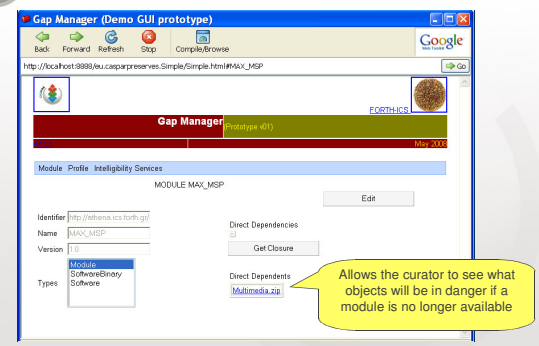
GWT-based prototype
List All Profiles



GWT-based prototype
Intelligibility-aware Packaging (DIPs)



GWT-based prototype



GapManager and DataHolders

Methodologies that could be (directly) adopted

[SIMPLE] Very simple and easy

- 1/ Each information object is recorded to gap manager
  - only its id and its name is required
- 2/ The dependencies of these objects are specified
  - dependencies may point to modules that are already recorded (more that 2200 modules already exist)
- 3/ Appropriate DC profiles are defined
  - Just selecting the modules that are assumed to be known

[Advanced]

- 0/ Identify Tasks => Dep types => Specifialization of the Core Ontology
- Then continue steps 1/ .. 3/ of the [Simple] methodology

Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval

thanks for your attention

backup slides

CASPAR Research on Knowledge Management

Outline

Research Results of the 1st year (in brief)

Research Results of the 2nd year: on Knowledge Evolution

- On Comparing RDF Knowledge Bases
- On Ontology Evolution
- On Archiving RDF Knowledge Bases

Component vs State of the Art

- Gap Manager
  - Enables Intelligibility-aware Services. Support of DC profiles, extensible typologies (of modules and dependency types), computation of closures and gaps, import/export in various formats
  - More rich and expressive functionality comparing to other registries
- SWKM
  - Complete and scalable suite of basic services for validating, storing, querying, updating and *exporting* ***descriptive metadata*** expressed in RDF/S.
  - All services are based on a common knowledge repository enabling the consistency of its contents.
  - Advanced evolution services (declarative update languages, comparison services, versioning, ontology evolution)

1st year Results

Contributions

- A formalization of intelligibility and intelligibility gap through the notion of dependency. This view could be used for providing answers to some basic questions (who much RI we need, how this depends on DC, etc)

- Y. Tzitzikas, "**Dependency Management for the Preservation of Digital Information**", 18th International Conference on Database and Expert Systems Applications, DEXA'2007, Regensburg, Germany, September 2007
- Y. Tzitzikas and G. Flouris, "**Mind the (Intelligibility) Gap**", 11th European Conference on Research and Advanced Technology for Digital Libraries, ECDL'2007, Budapest, Hungary, September 2007
- Yannis Tzitzikas, **Dependency Management for the Preservation of Digital Information**, International Conference PV'2007 (Ensuring the Long-Term Preservation and Value Adding to Scientific and Technical Data), Oberpfaffenhofen/Munich, Germany, October 2007

The implementation activities of the 2nd year have exploited these results

Motivation for Research on Knowledge Evolution and 2nd year Research Results

Motivation

- **Evolution** is a key concept because preservation is a **dynamic process** (world evolves, software/hardware evolves, metadata schemas evolve, digital objects evolve, community knowledge evolves).
- Digital objects have to be preserved not only against hardware and software technology changes, but also against changes in its Designated Community

Requirements

- Bulk metadata updates, Versioning metadata and ontologies, Ontology Evolution, Change log management, Comparison operators, etc

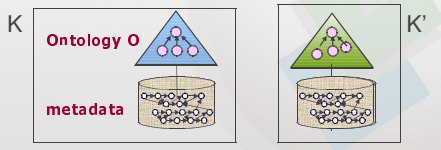
Research Results

- On Comparing Knowledge Bases
- On Ontology Evolution
- On Archiving

CASPAR RESEARCH RESULTS
Comparing RDF Knowledge Bases

- D. Zeginis, Y. Tzitzikas, V. Christophides, "**On The Foundations of Computing Deltas between RDF models** ", ISWC'2007 (Nov 2007)
  (**BEST PAPER AWARD!**)

Topic: What is the delta between two knowledge bases? Can we reduce the number of change operations needed to be applied on K so that to reach K'?



Delta(K->K') = ?

CASPAR RESEARCH RESULTS

Comparing RDF Knowledge Bases

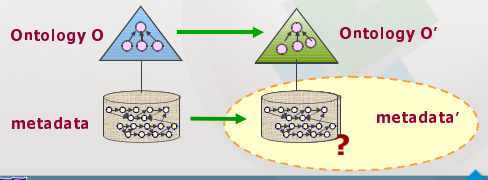
- Why it is useful in preservation
  - Consider the case where PDI is represented as an RDF/XML file (that instantiates one or more ontologies e.g. CIDOC CRM)
  - Suppose a new version of that file arrives
  - Issue: How the SWKM repository (that is used by Finding Aids) is updated?
  - Solution: Diff(file1, file2) gives a set of change operations that can be forwarded to SWKM to update it
- Research/Development Status
  - A prototype implementation over SWKM (as a Web Service) is ongoing.
  - Also a main memory implementation is already available

CASPAR RESEARCH RESULTS

Ontology Evolution

- G. Konstantinides and G. Flouris, G. Antoniou and V. Christophides, "**A Formal Approach for RDF/S Ontology Evolution**", ECAI'2008 (accepted for publication)

Topic: How to reflect ontology changes to the underlying data?



CASPAR RESEARCH RESULTS

Archiving Versions> Synopsis

- Y Tzitzikas, Y. Theoharis, D. Andreou, "**On Storage Policies for Semantic Web Repositories that Support Versioning**", European Semantic Web Conference 2008, ESWC'2008 (June, 2008)

- This is the first work that focuses on the storage aspect of SW repositories that support versioning.
- We proposed an index called **POI** (**P**artial **O**rder **I**ndex), we verified the space gains of this index experimentally and we provided an efficient version insertion algorithm with acceptable main memory space requirements.
- From our experiments, POI can be 180 times more space economical compared to **IC** (Individual **C**opies-approach) and 18 times compared to **CB** (**C**hange-**b**ased approach) for parallel version tracks. Moreover, POI allows performing efficiently various cross-version operations.
- POI is an advantageous approach for archiving set-based data (e.g. an RDF KB is a set of triples), especially good for inclusion-related and ``oscillating'' data

Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval

thanks for your attention

Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval

more backup slides

Exploiting DC profiles for having intelligibility aware AIPs and DIPs

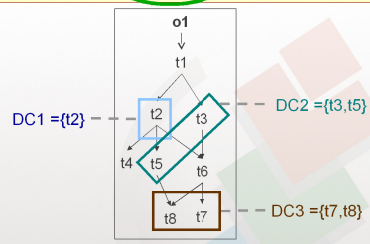
According to OAIS:

- **AIP** (Archival Information Package): It is actually a format which consist of
  - Data Object
  - Rep Info
  - PDI (**P**reservation **D**escription **I**nformation): e.g provenance, context, fixity
- **DIP** (Dissemination Information Package): Is the version of the information package delivered to the Consumer in response to an access request. May differ in form (e.g. TIFF to JPEG) or content (e.g. amount of metadata supplied) to that which resides in the archival store.

DC profiles could be exploited so that

- to be able to derive different AIPs and DIPs for different DCommunities
  - If the dependencies are available this could be done automatically

Exploiting DC Profiles for constructing the "right" DIPs (intelligible and redundancy free)



DIP of o1 wrt DC1

Object = o1
deps = {t1,t3}

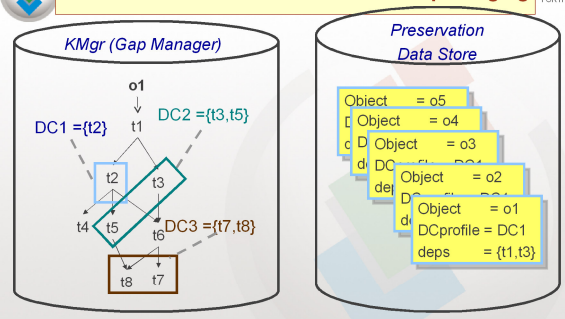
DIP of o1 wrt DC2

Object = o1
deps = {t1,t2,t4}

DIP of o1 wrt DC3

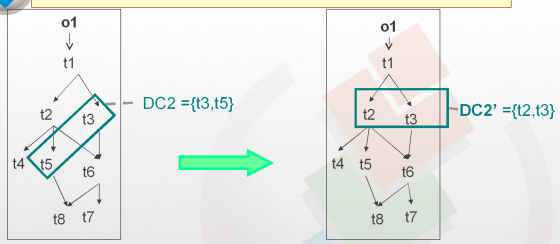
Object = o1
deps = {t1,t2,t3,t4,t5,t6}

Repackaging



And of course, the availability of dependencies and DC profiles could aid in transforming the stored AIPs to DIPs of different profiles (than those of the AIPs)

When DC profiles evolve



AIP of o1 wrt DC2

Object = o1
DCprofile = DC2
deps = {t1,t2,t4}

AIP of o1 wrt **DC2'**

Object = o1
DCprofile = DC2
deps = {t1}

Intelligibility-related dependencies and provenance