Autonomic Preservation of "Access Copies" of Digital Contents

Walter Allasia[1], Fabrizio Falchi[2], Francesco Gallo[1], Carlo Meghini[2]
[1]*EURIX S.r.l. walter.allasia@eurixgroup.com*
[2]*ISTI-CNR*

Abstract
The paper proposes the abstract approach enabling the delegation of the preservation processes that can be applied automatically to the access copies. The current standard, OAIS, is a reference model only and does not prescribe or even guide implementation. In particular, OAIS is designed around the Archival Information Package, leaving the Dissemination Information Package outside its boundaries. At the same time the "access copies" created as Dissemination Packages, must be preserved as well and very often can be considered as the cultural heritage the user is able to access. Small institutions as well as private users usually cannot afford a dedicated Archival Information System for preservation and their access copies often constitute the contents themselves. This paper investigates and analyses the problem of autonomic management of the "access copies" preservation. Autonomous agents have been conceptualized in order to manage the stored contents and to keep them updated. The representation of dependencies and obsolescence has been analysed for allowing the automatic application of migration paths.

Authors
Walter Allasia graduated in Physics in 1994 at the University of Turin with a thesis on the design of front-end electronic for nuclear detectors, tested at CERN, joined EURIX in 1997 and worked as software architect and designer of Enterprise Object Oriented Applications. Since 2001 he is Professor by contract at Physic University of Turin. He has been involved in several international projects such as PrestoSpace, PrestoPRIME, SAPIR, DMP and contributed to several MPEG standards such as MPQF, MAF, MPEG-M. During the PhD he was involved in setting up an Interoperable Digital Rights Management Infrastructure on a structured overlay network. Currently his research is focusing on Information Retrieval and Digital Preservation. Since March 2011 he is co-chairing the MPEG Ad Hoc Group on Multimedia Preservation.

Fabrizio Falchi is a researcher at ISTI-CNR in Pisa working on multimedia information retrieval, multimedia content management systems, content-based image retrieval, Peer-to-Peer systems. He received the MSc in Computer Engineering from the University of Pisa, a Ph.D. in Information Engineering from University of Pisa, and a Ph.D. in Informatics from Faculty of Informatics of Masaryk University of Brno. He has a MBA in "Innovation Management & Services Engineering" from Scuola Superiore Sant'Anna in Pisa. He has been involved in several national and international project including SAPIR, VISITO Tuscany, NeP4B, ASSETS.

Francesco Gallo graduated in Physics in 2001 and obtained a PhD in High Energy Physics in 2004 at University of Torino. From 2000 to 2006, as associate member of INFN Torino, he has participated in international experiments at Stanford Linear Accelerator Center, California. His research activity focused on the study and development of software algorithms and analysis tools for the precise measurement and validation of the Standard Model of fundamental interactions. Since 2006 he has been involved in several R&D projects in EURIX, such as SAPIR and PrestoPRIME, covering different research areas: information retrieval, digital preservation, digital media processing. Currently he is responsible for the development of the PrestoPRIME preservation platform and is editor of the ISO/IEC 21000-19 (MPEG-21 CEL) standard.

Carlo Meghini is a prime researcher at ISTI, working in the area of digital libraries and digital preservation. In the area of digital libraries, he has been involved in the DELOS Network of Excellence in Digital Libraries, contributing to the DELOS Reference Model for Digital Libraries; he participated in the FP6 Integrated Project BRICKS, aiming at developing a distributed Digital Library Management System, in the DL.org coordination action, and is involved in the making of Europeana since 2007, through the EDLnet, Europeana version 1.0, Europeana version 2.0 and ASSETS Best Practice Networks. In the area of digital preservation, he has been involved in the CASPAR project, an FP6 Integrated Project aiming at developing an OAIS-based architecture for preservation; he has also taught the OAIS Reference Model in several events organized by the CASPAR Project in conjunction with PLANETS and DPE Network of Excellence in Digital Preservation.

1. Introduction

Setting up an Archival Information System for digital preservation is a complex topic covering several aspects. The paper proposes to make use of Autonomic agents for setting up a preservation framework able to manage the digital "obsolescence" automatically. Focusing attention to a specific context, the access copies, the authors introduce the backbone for enabling autonomic preserving. Section 2 reports the state of the art and current technologies in the field of digital preservation and autonomic systems. Section 3 introduces the Autonomic approach discussed in this paper, how to discover obsolescence and how to manage it autonomically, such as for example how to identify automatically the best migration path for contents and formats according to their relationships and dependencies: a model for these descriptions is presented in Section 4. Finally section 5 gives a preliminary design of autonomic preservation, specifically addressing the "access copies". Section 6 provides an overview of some future work that can be already foreseen, some currently investigated by the authors.

2. Background

The contribution proposed by this paper is built on top of the OAIS model and the concept of autonomic computing, which are briefly introduced in the following. Open Archival Information System (OAIS) reference model [OAIS] was developed to standardize digital preservation practice and provides a set of recommendations for preservation program implementation. It uses Representation Information to transform a data object into an information object and the Archival Information Package (AIP) to enable collections of information to be preserved over time by identifying the metadata needed for preservation. The OAIS model has been adopted in several projects and initiatives as the basis for the design and implementation of digital preservation systems.

Introduced by Paul Horn at IBM in 2001 [Horn 2001], Autonomic Computing refers to self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users. In [Brazier et al. 2009] the relationship between autonomic computing and agents has been studied. In particular, knowledge and reasoning, planning and scheduling, and inter-agent communication developed for individual agent systems are especially appropriate for autonomic computing.

Several projects have been funded so far, focusing on the topics mentioned above. An exhaustive list is beyond the scope of the paper; the following projects and initiatives focused on digital preservation using different approaches and are worth mentioning:

- Preservation and Long-Term Access Through Networked Services (Planets)[1] was a project co-funded by the European Community under FP6 to build practical services and tools to help ensure long-term access to digital cultural and scientific assets.
- Cultural Artistic and Scientific knowledge for Preservation, Access and Retrieval (CASPAR)[2] was EU FP6 project that aimed at implementing, extending, and validating the OAIS reference model, enhance the techniques for capturing Representation Information and other preservation related information for content objects.
- Automated Obsolescence Notification System projects (AONS and AONSII)) [Curtis et al. 2007, Pearson 2007] have been software development projects by the National Library of Australia in conjunction with the Australian Partnership for Sustainable Repositories (APR). The projects aimed at developing a software tool that automatically finds and report indicators of obsolescence risks, to help repository managers decide if preservation action is needed. As reported in [Pearson et al. 2008], there is still a mismatch between this objective and the available sources of information on file formats and further development related to format obsolescence is needed.
- SCIence Data Infrastructure for Preservation - with focus on Earth Science (SCIPID-es)[3] is an EU FP7 project that aims at delivering generic services for science data preservation as part of the data infrastructure for e-science and to build on the experience of the ESA Earth Observation Long-term Data Preservation (LTDP) programme.
- Enabling kNowledge Sustainability, Usability and Recovery for Economic value (ENSURE)[4] is a research project funded by the European Community under FP7 Programme, to extend the state-of-the-art in digital preservation to heterogeneous data. The use cases adopted come from healthcare, clinical trials, and financial services.
- From Collect-All Archives to Community Memories (ARCOMEM)[5] aims at helping to transform archives into collective memories more tightly integrated with their community of users, exploiting Web 2.0 and the wisdom of crowds to make web archiving a more selective and meaning-based process.
- SCAlable Preservation Environments (SCAPE)[6] addresses scalability of large-scale digital preservation workflows aiming at enhancing the state-of-the-art developing infrastructure, providing a framework for automated, quality-assured preservation workflows, and integrating these components with a policy-based preservation planning and watch system.

---

[1] http://www.planets-project.eu/
[2] http://www.casparpreserves.eu/
[3] http://www.scidip-es.eu/
[4] http://ensure-fp7.ey/
[5] http://www.arcomem.eu/
[6] http://www.scape-project.eu/

- Digital Preservation for Timeless Business Processes and Services (TIMBUS)[7] explores the digital preservation problem in scenarios in which the important digital information to be preserved is the execution context within data is processed, analysed, transformed and rendered.
- Alliance Permanent Access to the Records of Science in Europe (APARSEN)[8] is a Network of Excellence that looks across the work in digital preservation, which is carried out in Europe and tries to bring it together under a common vision.
- PrestoPRIME[9] is a research project funded under the FP7 programme, aiming at developing solutions for the long-term preservation of audio-visual digital media objects, programmes and collections, and finding ways to increase access by integrating the media archives with European on-line digital libraries in a digital preservation framework. Tools and services will delivered through a networked CompetenceCentre[10]

The new directions in long-term digital preservation as covered by the ENSURE, ARCOMEM, SCAPE and TIMBUS EU projects have been recently discussed in [Edelstein et al. 2011]. The discussion underlines that ACROMEM stands alone in dealing with publicly available and non-regulated data while TIMBUS is the only one focusing on the environments. ENSURE and TIMBUS are both motivated in part by accurate risk assessment and preservation lifecycle issues related to regulations and together with SCAPE they also address the scalability of the infrastructures. Central to all of the stated projects is the ability to define what data needs to be preserved. The automation of the preservation lifecycle is being dealt with by the entire project except ARCOMEM. While SCAPE will be creating preservation lifecycles for deployment on large computational clusters, ENSURE and TIMBUS will examine how to extend existing lifecycle management tools to meet the additional requirements that digital preservation entails. Additionally, projects such as CASPAR and PrestoPRIME investigated the issues related to enabling future access, including representation and management of rights associated to digital content. In particular, PrestoPRIME developed a rights ontology model based on the analysis of narrative contracts in the B2B environment, which is currently under standardization as part 19 of MPEG-21 (ISO/IEC 21000-19). The PrestoCentre, the competence centre established by PrestoPRIME, aims at collecting best practices, guidelines and solutions at a European level, including support for standardization activities such as MPEG Multimedia Preservation Description Information, aiming at providing a standard interoperable preservation description that is capable of preserving multimedia content for long-term.

3. The Autonomic Approach for Digital Preservation

Starting from the OAIS [OAIS] model and from the experience matured in the projects cited in the previous section, we focus our attention to the access copies, going a little bit beyond the OAIS, introducing the autonomic management of preservation within the functional block "preservation planning", limited to access copies of the archive. According to [Kephart et al. 2003] we can image the "preservation planning" component made up of agents, autonomic agents linked together between federated networks of archives, able to perform two main tasks autonomously:

[7] http://www.timbusproject.net/
[8] http://www.alliancepermanentaccess.org/
[9] http://www.prestoprime.eu
[10] http://www.prestocentre.eu

discover the obsolescence of preserved contents
migrate the obsolete contents to the most appropriate format

Software already updates itself and the reader can easily recall the automatic software update notification for packages such as those provided by Adobe and OpenOffice. Actually these examples are dealing with specific software installed on specific hardware and last but not least, the software vendor with a specific registry updated for storing and publishing the obsolescence and the "update paths" to be followed.

Actually on the one hand these proprietary software packages are able to self-update, migrating automatically to the newer version, but on the other hand they have no knowledge about dependencies and consequences derived from the proposed update. Many times, the user is suffering problems from the self-update of software applications, because the operation breaks dependencies or, worse, updates some libraries shared with other software components that will never work again. Summing up, the "*self-update of software packages*" represents a menace for the user instead of being a benefit. Specific aspects to point out are:

- Self-update is usually not able to recognize and identify which other components are making use of the obsolete software that potentially cannot work anymore. As example, moving from Excel 2007 to Excel 2010 requires a spreadsheet format migration from xsl to xslx that is not always straightforward and maybe not working especially if file protection is used. Furthermore, if some application is specifically written for having "xls" files as input, it will never work again after the update.
- Self-update is not able to evaluate which software components have been used in the older versions, that are candidate to be broken and do not work anymore.

Hence, instead of simply asking the user "A new version of software X is available, do you want to install it?" it would be better if the self-update process was able to ask the following question: "A new version of software X is available, if you want to install it you have to migrate the following packages and in order to have these software component still running, you have to move these libraries to something else, etc.". Unfortunately as the reader has already experienced, the self-update of software packages is a serious issue and usually completely left to the user's responsibility, without any support and advice on potential damages that the process can cause.

In our case, the autonomic preservation of access copies, the problem is really much wider, without any well established, trusted and complete registry, managing and preserving not only a specific software version but also and mainly the contents and the technology and software components linked or with dependencies associated.

The automatic update of archived master copies is weakly perceived because the responsible archives are already setting up procedures and resources dedicated. Completely different situation is what we have about the access copies, for usually there are no many resources to allocate and an automatic support is welcome and needed.

Even if in Section 2 we have pointed out some experiences and projects addressing these tasks, anyone has come up with a complete solution. Actually most of them are addressing the general preservation issues; hence the loss of a single bit is not allowed for archived copies, resulting in the impossibility to trust any automatic system performing format migrations and conversions without the supervision of human beings. On the other hand, limiting the scope to the "access copies" (or "wife copies"), we can afford the loss of contents as well as the migration to a format chosen automatically by the agent that may be not the best. These faults can be accepted because there is always the master copies and therefore it is possible to re-create access copies at any time, thereby exposing again the original contents.

Moreover, focusing the attention to the "access copies" and to the "access" aspects, we can address also a further issue, going beyond OAIS, the preservation to the "publication system".

The "accessibility" to digital contents is not only limited to the contents but also the system providing the contents as well, connected to the clients with several different technologies involved.

We can figure out an "autonomic manager" responsible for the managed elements that are made up of our "access copies" together with the systems related to the access process (publication system, web servers, etc.). The "autonomic manager" is responsible for performing the following functions without the human intervention [Brazier et al. 2009]: self-configuration, self-healing, self-optimization, self-protection.

In our "preservation" context these functions can be implemented as follows:

Self-configuration means have a complete knowledge of preserved formats, especially if preserved contents are multimedia where we have wrappers/containers and different tracks for audio/video/text with associated codecs. Also the knowledge of all the "accessibility" aspects in charge to a specific node.

Monitor must also check what is going to be performed within the federated networks of other "autonomic agents", selected choices and migrations strategies applied. More specifically, in this phase the autonomic manager will evaluate configuration files, registries, makefiles, classpaths, shared and linked dynamic libraries, etc., generally every software and hardware structure having "usage" dependencies each other and within the architectural components. An agent must configure itself in order to receive the obsolescence events and selected the best actions to perform. Section 4 describes how to represent dependencies in order to set up the needed framework that the agent can apply in an autonomous way.

Self-protection means that the manager must proactively identify preservation issues and take appropriate actions. The "agent" receives messages about obsolescence risks and issues to be managed. Hence self-protection means that he will take the most appropriate actions in order to remove the issues.

Self-optimization means that once the autonomic manager has analysed a specific issue and decided to intervene for self-protection as well as self-healing, a ranking of the different, possible plans will be derived, taking into account efficiency and performance requirements, constraints and the rules to be followed. The ranking will be used to select the most convenient plan, either autonomously or by asking human intervention.

Self-healing is the action of autonomously evaluating the knowledge acquired in the monitor activity, in order to come up with suggestions and decisions to implement. This is the most important activity delegated to a preservation system: the healing analysis must decide the most appropriate action to be taken for preserving the access copies and the access system at risk. The following Section 4 describes how to represent the obsolescences in order to apply the rules that the agent uses for self-healing in an autonomous way from obsolescence. Self-healing represents the "cure" to the "obsolescence disease".

Having in mind the above concepts and taking into account next section (Section 4) describing the representation of obsolescence and dependencies, we can sum up and design the overall system (Section 5) able to autonomously manage the preservation of access copies.

4. A Model for Dependencies and Obsolescence Representation

Work on digital preservation so far has been largely inspired by the OAIS reference model [OAIS], which defines the information and the functional model of an archival system for the preservation of some information content. Although OAIS was never meant to be a design pattern, and even less a functional specification, it has been essentially interpreted that way. As a result, projects addressing digital preservation (from now on, DP for short) almost invariably end up designing *ad hoc* DP systems whose information structures and functional architectures reflect, respectively, the information and the functional models of OAIS. These DP systems are used as external entities, which are handed over the output of some other system, in order to preserve such output. In these approaches, preservation is achieved through a de-contextualization of the information to be preserved, and a great effort is required in order to endow the de-contextualized information with enough knowledge (Representation Information and PDI) to allow its interpretability in the future.

Contrary to these approaches, we think that the long-term preservation of information should be the **effect** of a **quality** that any information system can exhibit. We call such quality as *longevity*. Longevity is *the ability of an information system of achieving a set of goals that have a duration in time, no matter how distant in the future*. Whenever the duration in time of a goal extends beyond the lifetime of the technology used to achieve that goal, a DP problem arises. Thus, longevity captures DP as it is usually defined.

Current software development methodologies take a great care to guarantee that newly constructed information systems (from now on ISs, for short) satisfy a set of goals *at IS release time*. To this end, these methodologies provide languages for explicitly representing goals as requirements, and for relating requirements to the artefacts that document the software development process, from the design of the IS down to its final architecture. In the UML [ref], for instance, requirements are represented as use cases, and are related to the structural and behavioural elements that implement them. This relation is a *realization*: it associates use cases to collaborations, defined as "societies of classes, interfaces, and other elements that work together to provide some cooperative behaviour" [ref UML pg. 371]. Interfaces, in turn, are realized by components. Components are the basic elements of architectures, defined as "physical and replaceable parts of a system that conforms to and realizes a set of interfaces" [ref UML pg. 345]. By composing the realization relation, then, the UML allows to build a chain that connects requirements to the interfaces that realize them, and interfaces to the components that realize them. This chain gives therefore a very solid representation of the connection between the goals of an IS and the physical parts of the IS that achieve these goals, and as such it is crucial for longevity.

Unfortunately, OAIS has driven attention away from this chain, by proposing an Information Model that does not include any element of the chain. However, we argue that this chain is crucial for achieving a core functionality of OAIS, namely the "Develop Preservation Strategies and Standards" function, which is part of the Preservation Planning functional entity of the OAIS Functional Model. In fact, our proposal can be understood as directed towards supporting the development of preservation strategies through automation.

Any IS developed according to a principled methodology is thus born to *potentially* satisfy longevity. But in order to *effectively* satisfy longevity, an IS must successfully cope with the passage of time. The passage of time determines the *obsolescence* of the three fundamental elements of an IS:

- *Software components*: The obsolescence of software components is a well-known fact. It is caused by the perpetual evolution of technology and affects longevity due to the fact that the interfaces of obsolescent components will no longer be implemented, causing a set of goals to be no longer achieved.
- *Contents*: The obsolescence of the contents of an IS (understood as multimedia complex objects) is mostly caused by the degradation of media. As a result, some component using the content stored on the obsolescent media is no longer able to function, again causing a set of goals to be no longer achieved.
- *Metadata*: the obsolescence of metadata is due to the fact that the ontologies evolve: Some term in an ontology may fall out of usage, or may change its meaning, and new terms may come into usage. As a consequence, the metadata that use the out-dated or the changed terms, or that do not use the newly introduced terms, do not convey the same meaning as before to their users (be they software or humans in the designated community), and as such they may no longer be used for the same purpose.

In order to cope with obsolescence, the architecture on an IS needs to evolve. Evolving architectures is at the heart of digital preservation. It amounts to replace a set of components of the architecture, whether software, contents or metadata, with different ones so that a new architecture is obtained *that achieves all goals in place*. In the case of software components, evolution means replacement with one or more different software components in order to preserve functionality; in the case of contents, evolution means transformation from one format to another one in order to preserve information; in the case of metadata, evolution means re-writing according to a different ontology, in order to preserve meaning. The evolution of the architecture of an IS is intended to capture all these three different kinds of evolutions.

In this paper we do not consider the evolution of metadata, which requires techniques that significantly differ from those required for the evolution of software components and contents.

Most of the times the replacement of a component has consequences that impact other components of the architecture. Sometimes there is no unique replacement to be made, and a decision amongst a set of alternatives has to be taken, based on some utility function. Sometimes, it is not possible to know a priori whether an evolution of the current architecture satisfies all requirements in place, and a certain number of tests have to be executed to validate the evolution.

The approach

The methodology that we propose is based on the principle that the IS is endowed with, and uses an explicit representation of its own goals, of its own functional architecture, and of the relationships between the former and the latter that state how the goals are realized by the functions provided by the architecture. As already argued above, languages for expressing requirements and functional architectures already exist, and are currently used for software development. Those languages need to be extended by combining them with languages for representing goals, which have been researched in the last decade and are now at a mature stage of development [ref]. Indeed, goals stand at a more abstract level than requirements; moreover, they have their own structure and have a temporal dimension whose capturing is crucial for modelling and maintaining longevity. The methodology will also provide the machinery for representing the obsolescence of the IS architecture, as an event situated in time.

Depending on the type of the obsolescence, the corresponding event will carry contextual knowledge that will inform the ensuing evolution process. The methodology will finally provide the algorithms for analysing the obsolescence, determine which goals are affected by it, and propose a ranked set of alternative architectures, each representing a different way of evolving the current architecture in order to attain longevity of the IS. In order to compute alternative architectures, the methodology will need to acquire information about existing components for replacing the obsolescent ones.

Typically, an architecture can be evolved in many different ways, leading to a very large problem space. The methodology will deal with the possible combinatorial explosion by working on two different aspects: from the one hand, it will use a natural ordering amongst architectures ruling out architectures that are *a priori* sub-optimal, i.e. architectures that are functionally redundant, supersets of other architectures, and the like; from the other hand, it will rank architectures based on the combination of two main factors:

- An application-independent factor, measuring the *distance* between the current architecture and the alternative architecture. Research work is needed to analyse several distance measures in order to determine in an empirical way the one that best fits the use cases at hand.
- An application-dependent factor, measuring the cost of migrating to the new architecture from a domain specific point of view. The identification of application-dependent factors, requires a substantial investment in the analysis of specific domains. Since different domains will have different factors, the application-dependent function is best considered as an input to the methodology.

The final selection of the architecture will be carried out by a human user with the support of the methodology, which will provide this user with the information generated during the architecture ranking process.

5. Designing the Autonomic Preservation of Access Copies

We are integrating the basic concepts of Autonomic Computing, Multi Agent Systems (MAS) and Service Oriented Computing (SOC) [Frances at al 2009] in order to design a computing preservation system able to manage itself given high level preservation rules provided by the archival administrator. Actually our autonomic system is made up of individual preservation agents, responsible for managing several preservation aspects, most important being: knowledge, reasoning, planning and scheduling.

Knowledge is related to the “self-configuration” autonomic function (Section 3), where the agents acquire the competence on preserved contents, the associated metadata and the overall Information System (Section 4) in order to identify every link and dependency. It also takes into account rules set up by the archive administrator and the information coming from the connected “preservation agents” (or, as described later, from the “shared environment”) in order to have a complete awareness of the monitored context. **Reasoning** is mainly related to the “self-healing” autonomic function the focus point of the

preservation agent, implementing the capability to gather information from the knowledge in order to come up with appropriate preservation actions that the **Planning** will analyse and simulate in the preservation agent environment in order to be able to choose the correct procedures for Scheduling the list of tasks to be performed. As already pointed out, Reasoning is the core aspect of our preservation agent. The considered approach is the Stigmergy model, first introduced in [P. Grasse 1959], where preservation agents collaborate not by direct message exchanged but by jointly making and sensing changes to their shared environments and knowledge [Frances et al 2009][Babaoglu et al 2006][Blesson and McKee 2009]. In this way we are removing the need of a centralized registry of obsolescence (such as for formats) as we may have been forced to plan (especially in a SOC and SOA approach), moving the registry concept to the complete (and more comprehensive) view of the overall archival system, made up of Information Systems (software, hardware components), metadata and obviously "contents", representing the knowledge of the preservation agent. A typical scenario could be as follows:
Preservation agent A discovers a weak link in its knowledge, maybe a content format obsolescence, or an application for rendering a specific codec no more available or supported.
Preservation A reasoning, according to its rules, either:

- Suggests to migrate to a new format selected from an available list or based on a pre-existing knowledge; or
- Notifies the archive administrator in order to choose the most appropriate migration path to follow.

In the latter case the new rule is added to the knowledge of preservation agent A. Hence the preservation actions are applied to the contents/systems managed by preservation agent A (planning and execution). According to the Stigmergy model [Marsh and Onof 2009], preservation agent B is looking at the knowledge of its linked agents, including agent A. New rules are hence available addressing a specific format obsolescence and preservation agent B will reason on contents and systems which is responsible for. In the case B is suffering the same or a partial obsolescence issue, the preservation agent B will autonomously take the appropriate preservation action in order to fix it. It can decide to adopt a partial migration path in order to tailor the preservation action to its specific needs, rules and context. Again, a further preservation agent C in the network of A and B can watch what's happening and, even if it has no contents under obsolescence risk, he can update its knowledge for future use improving its reasoning capabilities addressing the overall migration path as well as sub-parts and specific weak links management.

The above scenario demonstrates that if on the one hand it is not possible to create a generalized obsolescence registry (as demonstrated by many initiatives (Section 2), on the other hand, making use of autonomic approach with Stigmergy preservation agents, it is possible to create customized registries (tailored specifically to preservation archives), that are distributed according to a shared knowledge managed by preservation agents.

6. Conclusions and future work

This paper has introduced a candidate solution for the preservation of digital contents, specifically addressing the "access copies". The approach has analysed and designed an autonomous model where autonomic systems are able to perform the preservation actions needed to keep contents (and their accessibility) up to date. A method for describing obsolescence has been presented and a solution based on stigmergy approach has been proposed. This position paper opens novel paths to follow enabling automatic and autonomic preservation.

In order to have a shared environment describing the knowledge of "autonomic agents" distributed over the federated networks, we can image to set up services published by Linked Open Data protocols. Making use of "graph" queries it could be possible to extract complex information, links and dependencies, that will be the knowledge base evaluated and modified in order to preserve the access copies. A software model based on the proposed approach can be designed and simulated on open source MAS (Multi Agent Systems) platforms such as Jade[11] and Janus[12], adding the needed graphs (or *RDF*13 *like*) for representing the knowledge base, continuously monitored and modified by the preservation agents. Currently the authors are investigating artificial life design patterns in order to select the most appropriate stigmergy model and set up a simulator for the experimentation of autonomic preservation processes.

Acknowledgements

This work was partially supported by the PrestoPRIME project, funded by the European Commission under ICT FP7 (Seventh Framework Programme, Contract No. 231161) and the Europeana v2.0, CIP-Thematic Network (contract No: 270902).

References

[OAIS] Consultative Committee for Space Data Systems. (2002). Recommendation for Space Data System Standards: Reference Model for an Open Archival Information System (OAIS). Blue Book

[Horn2001] Horn P: 'Autonomic computing: IBM's perspective on the state of information technology', also known as IBM's Autonomic Computing Manifesto, IBM (October 2001) -- http://www.research.ibm.com/autonomic/manifesto/autonomic_ computing.pdf

[Brazier et al. 2009] Frances M. T. Brazier, Jeffrey O. Kephart, H. Van Dyke Parunak, and Michael N. Huhns. 2009. Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda. IEEE Internet Computing 13, 3 (May 2009), pp 82-87.

[AONS] J. Curtis, P.Koerbin, P.Raftos, D.Berriman, J.Hunter, "AONS - An Obsolescence Detection and Notification Service for Web Archives and Digital Repositories", New Review on Hypermedia and Multimedia (JNRHM), January 2007.

[AONSII] David Pearson, "AONS II: continuing the trend towards preservation software "Nirvana". Paper presented at iPres2007, Beijing, China, October 11-12, 2007
[Pearson et al. 2008] Pearson, D., & Webb, C. (2008). Defining File Format Obsolescence: A Risky Journey. International Journal of Digital Curation, 3(1), pp 89-106. UKOLN.s

[11] http://jade.tilab.com/
[12] http://www.janus-project.org/Home
[13] Resource Description Framework - http://www.w3.org/RDF/

[Edelstein et al. 2011] Orit Edelstein, Michael Factor, Ross King, Thomas Risse, Eliot Salant, Philip Taylor, Evolving Domains, Problems and Solutions for Long-term Digital Preservation. Proc. iPRES 2011 - 8th International Conference on Preservation of Digital Objects.

[Kephart et al. 2003] Jeffrey O. Kephart, David M. Chess, The Vision of Autonomic Computing, *IBM Thomas J. Watson Research Center* 0018-9162/03 © 2003 IEEE

[P. Grasse 1959] P.P. Grassé, La reconstruction du nid et les Coordinations Inter-Individuelles chez Bellicositermes Natalensis et Cubitermes sp, Insectes Sociaux, Vol. 6, no 1 1959, pp. 41-84.

[Frances et al 2009] Frances M. T. Brazier, Jeffrey O. Kephart, H. Van Dyke Parunak, and Michael N. Huhns. 2009. Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda. IEEE Internet Computing 13, 3 (May 2009), 82-87.

[Babaoglu et al 2006] Ozalp Babaoglu, Geoffrey Canright, Andreas Deutsch, Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella, Niloy Ganguly, M\&\#225;rk Jelasity, Roberto Montemanni, Alberto Montresor, and Tore Urnes. 2006. Design patterns from biology for distributed computing. ACM Trans. Auton. Adapt. Syst. 1, 1 (September 2006), 26-66.

[Blesson and McKee 2009] Blesson Varghese and Gerard T. McKee, Applying Autonomic Computing Concepts to Parallel Computing using Intelligent Agents, World Academy of Science, Engineering and Technology 55 2009

[Marsh and Onof 2009] Leslie Marsh and Christian Onof. 2008. Stigmergic epistemology, stigmergic cognition. Cogn. Syst. Res. 9, 1-2 (March 2008), 136-149.