

ARTIST
FP7-317859



*Advanced software-based seRvice provisioning and
migrATion of legacy Software*

Deliverable D2.2.1

Standardization Report and follow-up roadmap

Editor(s):	Leire Orue-Echevarria Juncal Alonso
Responsible Partner:	TECNALIA
Status-Version:	Final – v1.0
Date:	15/03/2014
Distribution level (CO, PU):	PU

Project Number:	FP7-317859
Project Title:	ARTIST

Title of Deliverable:	Standardization Report and follow-up roadmap
Due Date of Delivery to the EC:	31/03/2014

Workpackage responsible for the Deliverable:	WP2 – Scientific and Technical Coordination
Editor(s):	Juncal Alonso (TECNALIA) Leire Orue-Echevarria (TECNALIA)
Contributor(s):	Juncal Alonso(TUWIEN), Leire Orue-Echevarria (TECNALIA), Philip Langer (TUWIEN), Martin Fleck (TUWIEN)
Reviewer(s):	Daniel Field (ATOS)
Approved by:	All Partners
Recommended/mandatory readers:	WP4 and WP3

Abstract:	This report will document all the standardization activities performed in the project from the beginning to the M18 month.
Keyword List:	OMG, CloudML, TOSCA, ReqIF, UML2, ATL, FUML, ISO CCRA
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/

Document Description

Document Revision History

<i>Version</i>	<i>Date</i>	<i>Modifications Introduced</i>	
		<i>Modification Reason</i>	<i>Modified by</i>
v0.1	10/01/14	First draft version	TECNALIA
v0.2	10/02/14	Integrated contributions from partners	TECNALIA
V1.0	15/03/14	Final version after revision	TECNALIA

Table of Contents

Table of Contents	4
Table of Tables	4
Terms and abbreviations.....	5
Executive Summary	6
1 Relevant Standards for ARTIST	7
2 Standardization strategy in ARTIST	12
3 Conclusion	14
4 References.....	15

Table of Tables

TABLE 1. USE OF THE RELEVANT STANDARDS FOR ARTIST	10
---	----

Terms and abbreviations

EC	European Commission
OMG	Object Modeling Group
UML	Unified Modeling language
CCRA	Cloud Computing Reference Architecture
ISO	International Standard Organisation
fUML	Foundational UML
GML	Goal Modeling Language

Executive Summary

The purpose of this document is to update the standardization strategy defined in M6, in D2.2. This is a key document wrapping the work being carried out in all technical workpackages.

The ARTIST standardization strategy has the following goals:

1. All ARTIST tools shall be based on standards in order to foster the uptake of the ARTIST solution by the industry. Members of the Special Interest Group (SIG) have expressed this requirement in order for ARTIST to have a wide impact in the industry.
2. Evaluate if any of the results can contribute to extend existing standards. Should this be the case, evaluate then how they can be extended and the procedure to follow
3. Evaluate if any of the results can have an impact as a new standard. Should this be the case, evaluate which standardization body is the most suitable one, and which procedures these bodies have.

The document is structured as follows. Section 1 analyses four new standards that have started to be used since M6 and updates the use of the standards analysed at that particular time. Section 2 outlines the standardization strategy and the identified activities to fulfil that strategy. Section 3 presents the conclusions and future steps.

1 Relevant Standards for ARTIST

This section will position the ARTIST work and strategy against the various concerned, existing and upcoming standards.

Quite a large number of standards are related to the ARTIST work that involves the following research fields:

- Software modelling and meta-modelling
- Cloud Computing
- Business modelling
- Reverse engineering
- Forward engineering
- Methodologies

For each of these fields, we can quote several standards. An initial set were presented in “D2.1 Plan for standardization” [1]. This document provides an update of the relevance for the selected as key standards in that phase of the project and presents a new survey of standards that have arisen, as potentially relevant for the scope of the project. The structure followed to outline these standards is similar to the one used in [1] which is:

1. Name of the standard (name, organisation name, organisation type (European, international, etc.))
 1. Name of the standard
 2. Goal and scope,
 3. Application to which phase and outcome of ARTIST,
 4. Extension activities within ARTIST, briefly explanation where this extension could be and of the processes explained above, which one we should follow

TOSCA [2]

Name of Standard: Topology and Orchestration Specification for Cloud Applications Version 1.0 (TOSCA), OASIS (International)

Goal and Scope: TOSCA focuses on the portability of cloud applications and services. TOSCA fosters interoperability among the services of an application, and the application with the infrastructure. To do so, it describes the services with templates across all layers of the as-a-service concept. Service templates describe the topology of complex services and applications. They can also specify the administration of such services and applications, e.g., deploy, patch, and scale out.

Application in ARTIST: TOSCA will be applied mainly for the definition of the technological maturity of an application. It is used in order to check whether the old and new application comply with the service concept as defined in the standard, and also the operational behaviour.

Extension in ARTIST: It is not envisioned to extend TOSCA, although CloudML, shall it become a standard, will be aligned with it.

ReqIF [3]

Name of Standard: ReqIF (Requirements Interchange Format), Object Management Group (International).

Goal and Scope: Having its origin in the automotive industry under the abbreviation RIF and standardized by the OMG as ReqIF in 2011, the Requirements Interchange Format provides a generic, non-proprietary XML-based format for exchanging requirements information between requirements authoring tools.

Basically, the format consists of a header with meta-information, and a collection of specifications, requirements (called SpecObjects) and relationships between those requirements. A ReqIF-compliant implementation must support all elements, as well as a high-level exchange protocol defined in the standard. Tool-specific information with no semantic counterpart in ReqIF, e.g., view information, can be attached using an extension element.

Application in ARTIST: Internally, ARTIST uses the ARTIST Goal Modeling Language (GML) to express migration goals, which can be seen as requirements, and connect those to other migration artifacts, such as code, transformations, and evaluations. To enable exporting and importing migration goals from and to existing requirements management tools, the ARTIST GML might be exported to a generic ReqIF-compliant XML document.

Extension in ARTIST: It is not envisioned to extend ReqIF, but to map our internally used ARTIST Goal Modeling Language (GML) to this format to promote interoperability.

fUML [4] [5]

Name of Standard: fUML (Foundational UML), Object Management Group (International).

Goal and Scope: The fUML (Foundational UML) standard formally defines the execution semantics of a subset of UML. This subset contains the most important modeling concepts for defining the structure of a system using UML classes and the behavior of a system using UML activities. The purpose of fUML is to serve as a computationally complete intermediate language between UML and a specific target platform language such as Java, enabling direct model execution. The fUML standard is accompanied by a Java-based reference implementation of a virtual machine, which can execute fUML-compliant models and computes the outputs from specified input parameter values.

Application in ARTIST: fUML is used to execute the migrated application on model-level and use the resulting runtime information of this execution to evaluate functional and non-functional migration goals (Task 11.1, 11.2, and 11.3). Using model-level execution in an early stage of the migration can help identifying unsatisfied migration goals before the actual code generation and deployment. Furthermore, fUML can be used to evaluate different deployment configurations or pattern (Task 9.5) based on their impact on the migration goals.

Extension in ARTIST: In ARTIST, we use an extended version of the reference implementation [6], which allows us to extract the required runtime information. Based on

this information, a framework is built to extract different structural and behavioral metrics necessary for the evaluation of the migration goals.

CCRA [7]

Name of Standard: CCRA Cloud Computing Reference Architecture – draft version, ISO (International)

Goal and Scope: The goal for developing CCRA is to enable the production of coherent set of international standards for Cloud Computing. The CCRA supports the following critical standardization objectives:

- enable the production of a coherent set of international standards for cloud computing
- provide a technology-neutral reference point for defining standards for cloud services
- encourage openness and transparency in the identification of cloud computing benefits and risks.

The CCRA is a way for describing, discussing, and developing a system-specific architecture using a common framework of reference.

CCRA supports a viewpoint approach to describe Cloud Computing systems:

1. User view: The user view addresses the activities, roles, parties and services.
2. Functional view: The functional view is a technology neutral view of the functions necessary to create a cloud computing system and it describes the distribution of functions necessary for the support of Cloud Computing activities.
3. Implementation view: The distribution of functions necessary for the implementation of a Cloud service
4. Deployment view: This view represents how the functions of a cloud service are technically implemented within already existing infrastructure elements
5. The implementation and deployment view are related to technology and vendor specific cloud computing implementations and actual deployments and are therefore out of scope of this Standard.

Application in ARTIST:

ISO CCRA will be used in the following tools:

- MAT, for all dimensions (architectural, business and processes). ISO CCRA covers aspects that range from cloud operational services (SLA, monitoring, security, policy ...), to business supporting ones (account management, product catalogue, subscription management, etc.).
- BFT in its Process-Kit tool, since ISO CCRA defines also a set of processes / activities for delivering a service.
- Methodology: ISO CCRA defines some processes that need to be considered in the business and process legs of the methodology.
- Certification model, in the technology axis.

Extension in ARTIST: It is not envisioned to extend the ISO CCRA or neither the associated standard that focuses on the cloud computing vocabulary.

In the deliverable D2.1, the ARTIST consortium analysed a set of standards. The following table shows in which tools these standards are being used

- PIM4Cloud, that branched into CloudML in 2013, after this project had started.
- SPEM2.0
- ITIL
- RAS
- KDM
- UML2
- MARTE
- ATL
- MOFM2T
- OCCI

Table 1. Use of the relevant standards for ARTIST

Standard	Tool	WP	Extending / using?
PIM4Cloud (*)	CloudML@ARTIST profile	WP7 and WP9	Extending, although now possibly part of TOSCA. Extension to be done in collaboration, if feasible, with MODAClouds and PaaSage
SPEM2.0	Methodology Process Tool (MPT), Methodology	WP6	Using
ITIL	MAT, BFT, SbSp certification model	WP11, WP5	Using
RAS	-	-	None
KDM	TFT, MDT	WP5, WP8	Using
UML2	TFT, MDT, MUT, Forward Engineering Integrated Environment, End-user based tests, behavioural equivalence tests, non - functional tests	WP5, WP8, WP9, WP11	Using
MARTE	Model-based simulation in the context of NFP computations (verification of non-functional tests)	WP11	Using
ATL	MDT, MUT, Forward Engineering Integrated Environment	WP8, WP9	Using
MOFM2T	-	-	None
OCCI	-	-	None

TOSCA	MAT	WP5	Using
ReqIF			None
fUML	model-level simulations of the migrated application to evaluate functional and non-functional migration goals (verification of functional tests)	WP11	Using
ISO CCRA			Using

(*) PIM4Cloud branched into CloudML in 2013, after this project had started

2 Standardization strategy in ARTIST

As stated in deliverable D2.1, Standardization has been identified as a key activity in ARTIST from the beginning of the project. As recommended by EC in [8] standardization has to be considered from the proposal phase with the idea that the earlier standardization is taking into account, the more benefit can be obtained from it. This strategy and report aims to follow this recommendation.

In the case of ARTIST, the goal of including a standardization task as part of the project is twofold. On one hand, the ARTIST consortium aims at creating a set of methods and tools that are adopted by the industry. In order to achieve that, these tools must be based on, or be compliant with existing standards, otherwise the uptake from the industry is not assured, minimizing the impact of the project results. On the other hand, the project aims at extending existing standards, if applicable and if feasible, with the research results attained in the context of the project.

The standardization activities will be led by TECNALIA, the technical coordinator of the project, which is an active research member with in-depth knowledge and experiences in initiating, driving and finalising standardization task forces. However, all partners of the project may participate in any of these activities. SPARX as member of the OMG is also actively participating in these activities.

In the previous deliverable, the consortium set up two main activities in the standardization strategy, being these 1) screening existing standards, 2) contribute to existing standards or develop new ones, 3) Identify results that can be standard-able

Activity 1: Monitor existing standards to include them, if relevant, in the ARTIST solution

Screening standards is an on-going task, which started at the kick-off of the project and will finish at the end of the project. This activity consists on the continuous monitoring of standards that are being published, either in a draft status or final status, which are relevant for the work being performed in the project. In the previous deliverable we catalogued these standards in several topics. The updated version of this categorization is shown next:

- Software modelling and meta-modelling: KDM,UML2,MARTE, fUML, CloudML
- Cloud Computing: CloudML, ISO CCRA
- Business modelling: ITIL, SPEM2.0, ISCO CCRA
- Reverse engineering: KDM,ATL,UML2
- Forward engineering: KDM,ATL

Activity 2: Contribute to on-going standardization activities

This activity started off at the beginning of the project and is currently going on. The goal of this activity is to develop new standards or contribute to existing ones by extending them. ARTIST's first aim is to use existing standards in its solution, secondly, extending existing ones if they do not cover the purposes of the ARTIST tools and thirdly, if a relevant result can be standardized, perform the necessary activities for getting them under a standard.

For the second goal (extending existing standards) it has been identified that CloudML could be extended. CloudML@ARTIST, that is, the CloudML dialect created in the context of our project needs, focuses mainly on the modelling of target providers (IaaS and PaaS) and on deployment scripts. The modelling of target providers was first started in PIM4Cloud but was centered into the modelling of IaaS and not PaaS. Additional features of IaaS were missing too.

In the context of ARTIST, PIM4Cloud has been extended and now covers more issues. For the case of the deployment scripts, ARTIST approach is at PIM level.

CloudML is also being used and extended in other European Projects such as MODAClouds and PaaSage. Both projects have created their own dialects. These two dialects plus the one created in ARTIST are in the process of being merged. Once this has been achieved, the possibility of creating a standard out of it will be further evaluated.

Another relevant issue to be able to extend existing standards is to participate in existing standardization bodies. These bodies often identify gaps that could be solved with projects like ARTIST. Partners of ARTIST are active members of OMG, CEN CENELEC and ISO CCRA JT/SC38.

Activity 3: Identification of potential standard able results

As stated in D2.2, ARTIST project will analyse and identify possible results to be part of a standard.

In the case of new standards, ARTIST does not have it as first aim. However, that does not imply that if the partners believe that a result could be standardized, that will not occur. For the time being, the Goal Modelling Language is closely observed as it has the potential of becoming a standard.

These decisions, the extension of existing standards and the creation of new ones, will be taken in Milestone 6 (after M30).

3 Conclusion

The main goal of this deliverable has been to update the standardisation strategy defined in D2.2. That deliverable identified several standards that were relevant for the ARTIST solution. Ever since, the project has been monitoring on one hand, new standards related to cloud computing that are emerging to include them in the solution, and on the other hand, the applicability of the standards identified in D2.2 in the final ARTIST solution.

Additionally, this document also presents the main three activities that are performed in this Follow-up task, which are the monitoring of standards, extension of existing ones with the ARTIST assets, and the analysis of which outcomes of ARTIST could be suggested as a potential standard.

4 References

- [1] ARTIST Consortium, «D2.1 Plan for standardization,» 2013.
- [2] OASIS, «Topology and Orchestration Specification for Cloud Applications Version 1.0,» 25 11 2013. [En línea]. Available: <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html>. [Último acceso: 29 1 2014].
- [3] Object Management Group (OMG), «Requirements Interchange Format (ReqIF),» OMG, October 2013.
- [4] Object Management Group (OMG), «Semantics Of A Foundational Subset For Executable UML Models (FUML), Version 1.1,» August 2013.
- [5] F. U. R. I. ModelDriven, «ModelDriven, Foundational UML Reference Implementation,» [En línea]. Available: <http://portal.modeldriven.org/project/foundationalUML>. [Último acceso: 10th February 2014].
- [6] P. L. a. G. K. T. Mayerhofer, «A Runtime Model for fUML.,» de *Proc. of the 7th Workshop on Models@run.time (MRT) @ MoDELS, pages 53–58. ACM., 2012.*
- [7] ISO/IEC JTC 1/SC 38 Distributed application platforms and services (DAPS), «ISO/IEC JTC 1/SC 38 Distributed application platforms and services (DAPS),» 2012. [En línea]. Available: <http://isotc.iso.org/livelink/livelink?func=ll&objId=15838397&objAction=Open>. [Último acceso: 24th February 2014].
- [8] CENELEC, «Integrating Standards in your FP7 project,» 2012.