EUROPEAN SPACE AGENCY
DIRECTORATE OF OPERATIONS AND INFRASTRUCTURE
OPS-GI

ROUND TRIP ENGINEERING

EXECUTIVE SUMMARY

Reference: egos-rte-rte-ES-0001
Version: 1.0
Date: 2008-08-08

EUROPEAN SPACE AGENCY
DIRECTORATE OF OPERATIONS AND INFRASTRUCTURE
OPS-GI

Document Title:	Round Trip Engineering		
Document Reference:	egos-rte-rte-ES-0001-version1r0+executive-summary		
Document Version:	1.0	**Date:**	2008-08-08
Abstract			

Approval Table:

Action	Name	Function	Signature	Date
Prepared by:				YYYY-MM-DD
Verified by:				YYYY-MM-DD
Approved by:				YYYY-MM-DD

Authors and Contributors:

Name	Contact	Description	Date
Alessandra Nanni	ali@terma.com	Author	2008-06-06
Damien Tamburri	Damien.tamburri@di.univaq.it	Author	2008-06-06

Distribution List:

Document Change Log

Issue	Date	Description
1.0	2008-08-08	Creation Date of the Document

Document Change Record

DCR No:	00	**Originator:**	
Date:		**Approved by:**	
Document Title:	Round Trip Engineering		
Document Reference:	egos-rte-rte-ES-0001-version1r0+executive-summary		
Page	**Paragraph**	**Reason for Change**	

TABLE OF CONTENTS

TABLE OF FIGURES

1. Introduction

The maintenance of legacy software represents a major investment for any organization. MDA Model Driven Architecture provides a mean to isolate the business and application logic form specific platform technologies where the program runs. The three primary goals of MDA are portability, interoperability and reusability. MDA is supported by a number of recognized standards as UML Unified Modelling Language and the OMG Object Management Group.

The project has been divided into 2 parallel Studies. Both studies investigate the current state of available technologies that can be applied to the MDA KDM Knowledge Discovery Meta-Model applied to the FARC Case Study. The main difference of the two studies consist in the fact that the current study, RTE Round Trip Engineering, is using a tool chain available on the market while the second study involved the tuning and customization of an existing MDA tool by a vendor.

The motivation for the study is related to the current implementation of EGOS which aims in standardizing the infrastructure through ESA ground segments. The goals of EGOS involve improving reliability, cost effectiveness and interoperability of the existing systems. For this reason ESA/ESOC is investigating new methodologies for reusing the knowledge within the legacy systems in order to save costs and leverage operational proofed software experience.

1.1 Purpose

The purpose of the executive summary is to provide an overview of the key points concerning the Round Trip Engineering Study.

1.2 Scope

The scope of the executive summary is related to the Reverse Engineering Methodology applied to the FARC File ARChive Case Study.

1.3 Glossary

1.3.1 Acronyms

Acronyms	Description
CIM	Computational Independent Model
CSV	Comma Separated Value
EGOS	ESA/EGOS Ground Operation System
EGOS-MF	EGOS-Modelling Framework
FARC	File ARChive
ISM	Implementation Specific Model
KDM	Knowledge Discovery Meta-Model
MDA	Model Driven Architecture
MOF	Meta-Object Facility
OCL	Object Constraint Language
OMG	Object Management Group
PDM	Platform Definition Model
PIM	Platform Independent Model
PSI	Platform Specific Implementation
PSM	Platform Specific Model
QVT	Query View Transformation
REM	Reverse Engineering Model
REQM	Reverse Engineering Quality Model
RTE	Round Trip Engineering

SoC	Separation of Concern
SUT	System Under Test
TN	Technical Note
TTCN-3	Test and Test Control Notation- version 3
U2TP	The UML Testing Profile
UML	Unified Modelling Language
XMI	XML Metadata Interchange
XML	Extensible Markup Language

1.3.2 Definition of Terms

Terms	Description
EGOS or ESA/EGOS Ground Operation System	The EGOS system presents a single architecture that divides the ESA ground segment data systems infrastructure into its high level components and the connections between them. These interfaces follow agreed international standards in order to have an application driven view.
EGOS-Modelling Framework	Formalizes frameworks for modelling space systems
Implementation Specific Model	The implementation Specific model is a low level model that is tightly linked to the code. It can be considered as the UML representation of the code. Within this model the level of detail information is the same as the one in the code.
Mapping	Specification of a mechanism for transforming the elements of a model conforming to a particular meta-model into elements of another model that conforms to another (possibly the same) meta-model. A mapping may be expressed as associations, constraints, and rules, templates with parameters that must be assigned during the mapping or other forms yet to be determined.
Methodology	A body of methods, rules, and postulates employed by a discipline.
Model	A model is a representation of a part of the function, structure and/or behaviour of an application or system. A representation is said to be formal when it is based on a language that has a well-defined form ("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described (e.g. message sends and replies, object states and state changes, etc.), or by translating higher-level language constructs into other constructs that have a well-defined meaning. The optional rules of inference define what unstated properties you can deduce from the explicit statements in the model. In MDA, a representation that is not formal in this sense is not a model. Thus, a diagram with boxes and lines and arrows that is not supported by a definition of the meaning of a box, and the meaning of a line and of an arrow is not a model—it is just an informal diagram.
Model Based Testing	Model-based testing is software testing in which testcases are derived in whole or in part from a model that describes some (usually functional) aspects of the system under test (SUT).
Model Driven Architecture	The MDA approach defines system functionality using a platform-independent model (PIM) using an appropriate domain-specific language. Then, given a platform definition model (PDM) corresponding to CORBA, .NET, the Web, etc., the PIM is translated to one or more platform-specific models (PSMs) that can run PSI. The PSM may use different Domain Specific Languages, or a General Purpose Language like Java, C++, PHP, Python, etc. Automated tools generally perform this translation such as QVT.
Object Constraint Language	The Object Constraint Language is a declarative language for describing rules that apply to UML models developed at IBM and now part of the UML standard. Initially OCL was only a formal specification language extension to UML. OCL may now be used with any Meta-Object Facility OMG meta-model, including UML. The Object Constraint Language is a precise text language that provides constraint and object query expressions on any Meta-Object Facility model or meta-model that cannot otherwise be expressed by diagrammatic notation. OCL is a key component of the new OMG standard recommendation for transforming models, the QVT specification.
Platform	A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented. [MDA]
Platform Definition Model	This is the intermediate step that allows models transformation from PIM to PSM. The PDM contains the information for the mapping between the high level PIM and the low level PSM.
Platform Independent Model	A platform-independent model or PIM is a model of a software or business system that is independent of the specific technological platform used to implement it.
Platform Specific Implementation	Is the specific implementation or code of a given platform
Platform Specific Model	A platform-specific model is a model of a software or business system that is linked to a specific technological platform (e.g. a specific programming language, operating system or database). Platform-specific models are indispensable for the actual implementation of a system.
Quality Model	Definition of what "quality of source code" means within a specific domain.
Query View	Is a model transformation language based on an expression over a view of a source model that is

Terms	Description
Transformation	transformed into target model.
Round Trip Engineering	An ESA Study on Model Driven methodologies applied to legacy systems.
Separation of Concern	is the process of breaking a design model into distinct features that overlap in functionality as little as possible. Typically, concerns are synonymous with features or behaviours. Progress towards SoC is achieved through modularity and encapsulation, with the help of information hiding.

1.4 References

1.4.1 Applicable documents

Ref.	Document Title	Issue and Revision, Date
[BSSC]	"Tailoring of ECSS Software Engineering Standards for Ground Segment in ESA, part A", BSSC (2005)1A. "Tailoring of ECSS Software Engineering Standards for Ground Segment in ESA, part B", BSSC (2005)1B. "Tailoring of ECSS Software Engineering Standards for Ground Segment in ESA, part C", BSSC (2005)1C.	1.0, 01/06-2005

1.4.2 Reference documents

Ref.	Document Title	Issue and Revision, Date
[BEL05]	Belinfante, A., Frantzen, L., Schallhart, C., Tools for Test Case Generation., in M. e. a. Broy, ed., Model-Based Testing of Reactive Systems, Springer LNCS, 2005, pp. 391–438.	-
[CH2007]	K. Czarnecki, S. Helsen, Feature-based survey of model transformation approaches, IBM Systems Journal, Volume 45, Number 3, 2006	-
[DF93]	Dick, J., Faivre, A., Automating the Generation and Sequencing of Test Cases, from Model-Based Specifications, Springer-Verlag, 1993.	-
[EGOS]	"EGOS LLC, Generic Specification Framework Guideline", EGOS-GEN-GEN-TN-1000.	2.1, 04/03-2006.
[EGOS-MF]	EGOSMF Specification - Issue 2.0.pdf	Issue 1.1 February 28, 2007
[EGOS-TN2]	"EGOS LLC, System Analysis", EGOS-GEN-GEN-TN-1001	1.3, 02/05-2006
[EGOS-TN4]	"EGOS LLC, Reusability Analysis", EGOS-GEN-GEN-TN-1004	1.2, 23/03-2007
[HAR02]	A. Hartman. Model Based Test Generation Tools. Technical report, AGEDIS project Downloads, 2002.	
[K2006]	I.Kurtev, K. van den Berg, F. Jouault, Evaluation of rule-based modularization in model transformation languages illustrated with ATL, Procs of the 2006 ACM Symposium on Applied computing, Dijon (France), ACM Press pp. 1202 – 1209, 2006	-
[MDA]	"MDA Guide Version 1.0.1", omg/2003-06-1, http://www.omg.org/docs/omg/03-06-01.pdf	1.0.1, 12/06-2003
[MDA-FAQ]	http://www.omg.org/mda/faq_mda.htm	-
[MT2006]	Procs of Model Transformation Track at ACM Symposium on Applied Computing (Eds: J. Bezivin, A. Pierantonio, A. Vallecillo), ACM Press. Dijon France, April 2006	-
[MTOOS06]	Leila Naslavsky, Henry Muccini and Debra Richardson. Scenario-based and State Machine-based Testing: An Evaluation of Automated Approaches. In M-TOOS 2006, First International Workshop on Model-Based Testing and Object-Oriented Systems. Affiliated with OOPSLA 2006.	-
[OMG]	www.omg.org	-
[PRE05]	Prasanna, M., Sivanandam,S.N., Venkatesan,R., Sundarrajan, R., A Survey on Automatic Test Case Generation, Academic Open Internet Journal, 2005.	-

Ref.	Document Title	Issue and Revision, Date
[PRE06]	Utting, M., Pretschner, A., Legeard, B., A Taxonomy of Model-Based Testing, Department of Computer Science, The University of Waikato, New Zealand, 2006.	-
[QVT]	"CBOP/DSTC/IBM: MOF Query/View/Transformations", 2nd Revised Submission, OMG ad/04-01-06.	-
[RFWC2004]	C. Raistrick, P. Francis, J. Wright, C.Carter, I. Wilkie. Model Driven Architecture with Executable UML. Cambridge University Press. 2004	-
[RTE-CIG]	Configuration and Installation Guide	1.1, 2008-04-08
[RTE-Doc]	FARC Documentation	1.1 2008-04-08
[RTE-SVTD]	Software Validation and Testing Document, Egos-rte-rte-svtd-0001	1.0, 2008-04-08
[RTE-TestReport]	RTE Test Report client-farc-c++-ttcn-report-001.pdf and client-farc-java-ttcn-report-001.pdf	1.0, 2008-04-08
[RTE-TN1]	"Technical Note Reverse Engineering Methodology",egos-rte-rte-tn-0002	1.5 2008-05-30
[RTE-TN2]	"Roundtrip Engineering Tool Chain Evaluation", egos-rte-rte-tn-0002	1.2, 2008-04-08
[RTE-TN3]	"Roundtrip Engineering Quality Model", egos-rte-rte-tn-0003	1.2, 2008-05-30
[RTE-TN4]	"Roundtrip Engineering Case Study", egos-rte-rte-tn-0004	1.4, 2008-05-30
[SCP2007]	Special Issue on Model Transformation, Journal of Science of Computer Programming (Eds: J. Gray, A. Pierantonio, B. Selic, A. Vallecillo), Elsevier	-
[TA04]	Bertolino A., Marchetti E., Muccini H. "Introducing a reasonably complete and coherent approach for model based testing", In Tools and Algorithms for the Construction and Analysis of Systems (TACoS'2004).	-
[TE05]	Zhen Ru Dai and Jens Grabowski and Helmut Neukirchen and Holger Pals. From Design to Test with UML -- Applied to a Roaming Algorithm for Bluetooth Devices. In Proceedings of the 16th IFIP International Conference on Testing of Communicating Systems (TestCom2004), Oxford, United Kingdom, March 2004. Lecture Notes in Computer Science 2978.	-
[TOGAF]	"The Open Group Architecture Framework – Enterprise Edition",	8.1, 19/12-2003
[U2TP]	OMG: UML 2.0 Testing Profile. Final Adopted Specification, ptc/04-04-02, 2004.	-

1.5 Document Overview

The main section in this document is:

Section 2: Executive Summary; This section provides highlights of the Round Trip Engineering Study, including aims/objectives, overall approach, findings, achievements, and conclusions.

2. Executive Summary

2.1 Background

ESOC is currently in a transition phase; from the legacy systems, to EGOS. In this context legacy systems are defined as 'any existing, operational software, regardless of the platform it runs on, language it is written in or length of time it has been in operation'.

There is therefore a need to be able to understand and reverse engineer software assets for the purpose of documentation, improvement, modification, interoperability, porting, migrations, reuse, redesign and/or redeployment. Previous work assessed the industrial readiness of MDA technologies and provided insight in their full potential. This work is intended in harnessing that potential with the goal of delivering a full-fledged round-trip solution, not only for the modernization of deployed legacy systems in ESA, but also of their harmonization.

2.2 Goals

The main objective of the study is to demonstrate the validity of the Model Driven Architecture approach for re-engineering.

2.3 MDA Methodology and Case Study

MDA represents an approach to system engineering where models are used as the central part for the maintenance and modification of a system. MDA promotes the idea of designing software systems at a platform-independent models (PIM) level which can be transformed to software implementation with model transformation technologies that incorporate the knowledge of the execution platforms in question. A PIM can be retargeted to different platform or PSM Platform Specific Model. Within this Study the FARC FileARChive legacy system has been used as case study and realized in a new platform target which is EGOS.

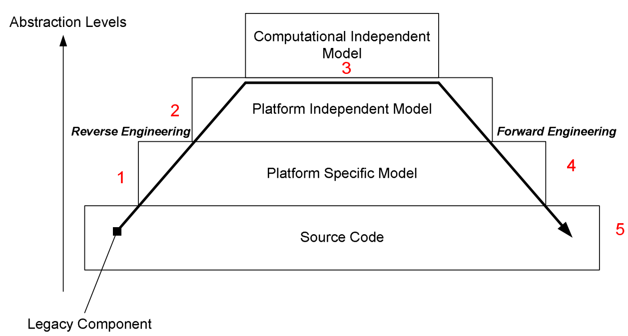

Figure 2-1 Model Driven Architecture

The Round Trip Engineering study focused on the following main phases:

1. Architectural Recovery where the code is produced into working diagram through reverse engineering and marked with a specific (methodology critical) profile according to the MDA foundation principle, SoC Separation of Concerns. Every class is given a role according to its overall function in the system and being refactored accordingly. This phase has been called REM Reverse Engineering Model and provides the PSM Platform Specific Model

2. PIM Platform Independent Model extraction focused on increasing the level of abstraction of the models by capturing the systems' high level platform independent view. The Architecture design will be provided in form of component and connectors, each in a black box fashion. Through this approach the details have been removed from the model.

3. The CIM Computational Independent Model of the system is imported form the legacy format. The CIM describes the enterprise viewpoint, such as requirements, constraints and use cases. The PIM and CIM links are satisfied through the use of stereotypes and profiles.

4. The Platform Independent Model is there after forward engineered, based on a platform model of EGOS system. The end result is an EGOS component with a lower level of abstraction of the PIM.

5. The last step in forward generation is to publish the model thus obtained PSM Platform Specific Model in a working PSM (or ISM Implementation Specific Model), synchronized with the code. Additional refactoring can be carried out on this model, including Design-Patterns installation, class constraining through OCL (Object Constraint Language).

The picture below provides an overview of the Platform Independent Model.

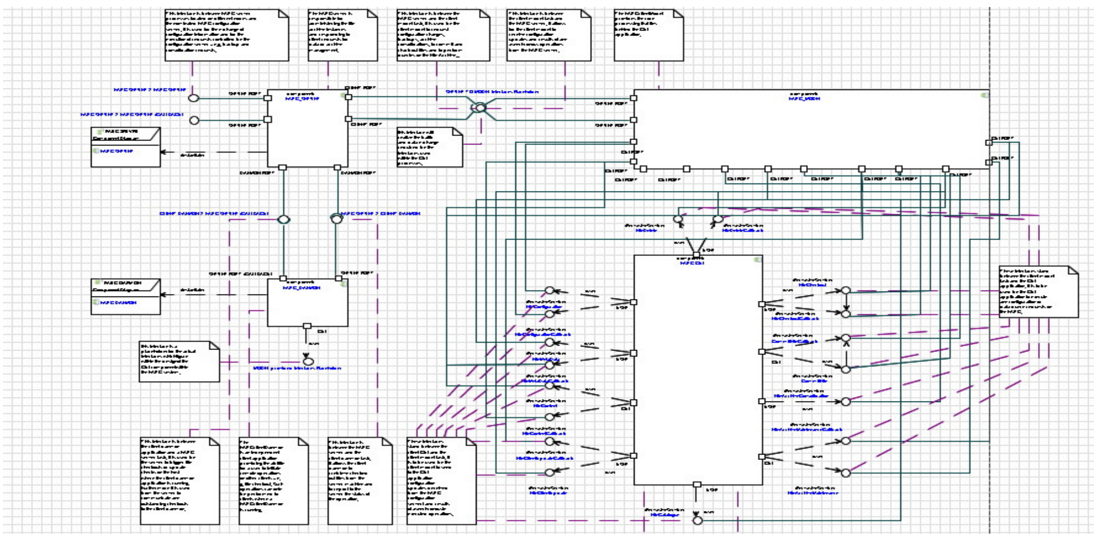

Figure 2-2 Platform Independent Model

Borland Together has been used as part of the Tool chain. It provided MDA tools such as OCL and QVT environment and Profiling Support. It also provided a UML modelling environment, an IDL and code generation support.

2.4 Model Driven Testing

The Testing Approach reuses the PSM and transforms it into a PST Platform Specific Test using the PIT Platform Independent Test. The models were marked with UML Testing Profile (**UTP**) and the behaviour of the test cases was defined through the use of sequence diagrams. UTP bridges the gap between the designers and the testers of a component, by allowing UML to be used for both system modelling and test specification. This allows a reuse of UML design documents for testing and enables test development in an early system development phase, and integrates the test suite closely with the model. The testcases models were automatically generated into TTCN-3 scripts by TTModeler and executed with TTworkbench. TTmodeler is an eclipse plug-in that can use UML models created by Topcased, Rational Architect Software and Enterprise Architect and transform them into TTCN-3 scripts using UTP Profile. TTModeler is an alpha tool provided by Testing Tech for the scope of this study.

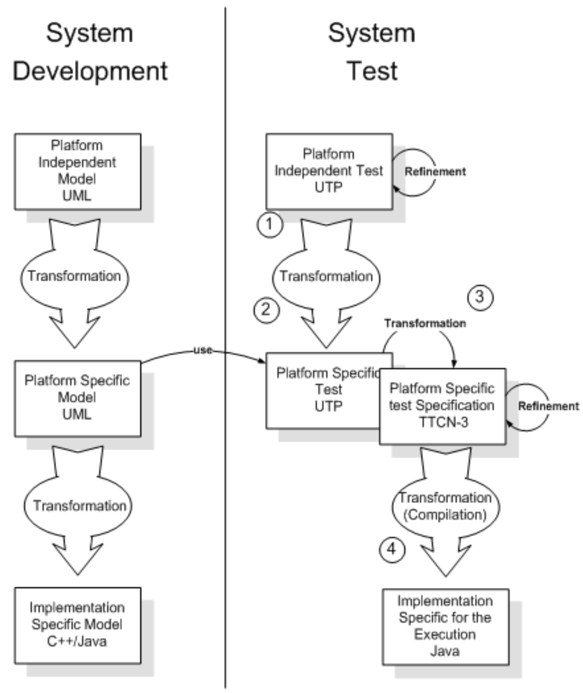

Figure 2-3 MDA and Test Based Model

2.5 Outcomes and Results

In its overall development, RTE project produced a number of valuable, the most important of which consist in:

1. The deployment of two working versions of FARC File ARChive, in an EGOS-Harmonized format. These two re-engineered versions in Java and C++ carryout most if not all the functions of the legacy FARC plus some additional behaviour.
2. The provision of a model-based testing methodology for the defined renewal process.
3. The provision of an effective way to EGOS Harmonize legacy systems.
4. The provision of a methodology, which allows shifting from traditional document driven development to model driven development.
5. The provision of an effective way to implement Separation Of Concerns during software development in mission-critical ESA systems.

The aims of focusing on a Model Centric Approach and of EGOS-harmonization of the Case Study were successfully accomplished and delivered as part of the project. RTE demonstrates how to shift paradigm focus from Document to Model driven development while also providing the methodology towards legacy renewal and EGOS harmonization.

2.6 Conclusions and Recommendations

Some final considerations may be drawn from the experience of the Round Trip Engineering Study. The project was in its overall aim intended at assessing the industrial readiness of model driven technologies towards legacy renewal. RTE proved that, while possible, legacy renewal is to be promoted, even with the proposed methodology, only in systems' whose size and increased number of functionalities renders it unfeasible to rebuild the system from scratch. Empirically, adopting the methodology is more and more “convenient” with respect to re-building the system from scratch, the more the systems' intended functionalities and size increases. It is still unclear whether other approaches to architectural recovery are effective in case of large and complex legacy systems such as those at ESA but the study made it clear that the level of automation could have been rendered high at will, depending on the investment.

The value for the whole RTE process is more evident when applied for the renewal of very big systems. It should be pointed out that this methodology involves a lot of technical and conceptual knowledge, for which a ramp up time has never to be underestimated. Moreover the tailoring of the required transformations, in order to apply them to new analyzed and then renewed systems is a very time consuming activity, for this reasons a trade off analysis has to be performed against a completely hand based rework of the same system.

It is important to highlight that most of the involved technologies are still under development and early evolution phases, thus the whole process cannot be supported in a completely automated optic, and some manual refinement has to be performed. Despite all this, the approach still automates a lot of the involved work, and it is also valuable to get a better understanding of the system, and thus is also a valid support in the very early phase of a renewal project. Another important notice is that there is still a good margin of improvement for the automation level of the whole process; new paths have to be investigated, and different solutions have to be taken into account, since the RTE has only scratched the surface of a really wide new methodology.

A fully fledged MDA approach is even more valuable for systems to be implemented, and it's there that MDA expresses its complete potential, since in such a development environment a higher level of automation can be achieved so far, not having to cope with non MDA-like choices, which are present in a legacy system.

The automatic generation of the behaviour is still very challenging and requires a large investment. The dynamic behaviour of a system can be captured in practice only in part. Source Code and model synchronization is straight forward in java but requires extra effort for C++.

Moreover the use of a tool chain instead of a single tool highlights the immaturity of the UML interchange standard. OMG is now working to provide a stricter standard on UML and MDA in order to support full tool interoperability within the near future.

Moreover a constant market scan should be performed, since some of the involved technologies are still in the early stages, and thus improvements and a better cohesion could be achieved in the future, supported by newer versions of the adopted tools or by completely different tools.

2.7 Acknowledgments

The team and ESA would like to thank all the technical staff at Borland Together and TestingTech. A special thank is especially addressed to the “Computer Science of University' dell' Aquila” (Italy) for their valuable co-operation and work in the development of the Round Trip Engineering Study.