European Space Agency
Directorate of Technical and Operational Support

Appendix 1 to AO/1-6138/09/NL/JK

STATEMENT OF WORK

Verification of Failure Impact by Model Checking

Reference: TEC-SWE/09-259/YY

Issue: 1.0 Revision 1

05.08.2009

European Space Agency
Agence spatiale européenne

ESTEC

Keplerlaan 1, PO Box 299, 2200 AG Noordwijk zh,
The Netherlands
Tel: +31 71 565 6565 Fax: +31 71 565 5060

Table of Content

1 Introduction

1.1 Scope of the Document

This document describes the activity to be executed and the deliverables required by the European Space Agency in relation to the research and development activity *Verification of Failure Impact by Model Checking*.

It will be part of the contract and shall serve as an applicable document throughout the execution of the work, with amendments as agreed at the kick-off meeting, if appropriate. It is organized as follows. Section 2 presents the background and the objectives of the activity. Section 3 presents in more detail the execution of the activity in providing a detailed description of the tasks. Section 4 lists management requirements and deliverables. Section 5 specifies schedule and milestones. To complete this document, Annexes A and B respectively provide the technical background of this study and the tailoring of the ECSS Software Standard (ECSS-E-ST-40C).

1.2 Applicable and Reference Documents

1.2.1 Applicable Documents (ADs)

The following documents, listed in order of precedence, contain requirements applicable to the activity:

[E-40C] ECSS-E-ST-40C – Space engineering – Software, European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 6 March 2009 (Tailored version, provided in the Annex B, is applicable)
ftp://escies.org/ecss.nl/ISO/ECSS-CD(6March2009).iso

1.2.2 Reference Documents (RDs)

The following documents can be consulted by the Contractor as they contain relevant information:

ECSS Standards

[P-001B] ECSS-P001B – Glossary of terms, European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 14 July 2004. http://www.ecss.nl/

[Q-30C] ECSS-Q-ST-30C – Space product assurance – Dependability, European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 6 March 2009. http://www.ecss.nl/

[Q-30-02C] ECSS-Q-ST-30-02C – Space product assurance – Failure modes, effects (and criticality) analysis (FMEA/FMECA), European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 6 March 2009. http://www.ecss.nl/

[Q-40C] ECSS-Q-ST-40C – Space product assurance – Safety, European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 6 March 2009. http://www.ecss.nl/

[Q-40-02C] ECSS-Q-ST-40-02C – Space product assurance – Hazard analysis, European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 15 November 2008. http://www.ecss.nl/

[Q-80C] ECSS-Q-ST-80C – Space product assurance – Software product assurance, European Cooperation for Space Standardization (ECSS), ESA Publications, Noordwijk, The Netherlands, 6 March 2009. http://www.ecss.nl/

ESA Studies

[OBMC] On-Board Model Checking – Autonomous Reasoning Engine, Final Report, Issue 1, ESA Contract 20580/07/NL/JD, 16 December 2008

Other Documents

[Ale07] B.J.M. Ale, L.J. Bellamy, R. van der Boom, J. Cooper, R.M. Cooke, L.H.J. Goossens, A.R. Hale, D. Kurowicka, O. Morales-Napoles, A.L.C. Roelen, J. Spouge, Further development of a Causal model for Air Transport Safety (CATS); building the mathematical heart, Risk, Reliability and Societal Safety – Aven & Vinnem (eds), 2007 Taylor & Francis Group, London http://dutiosc.twi.tudelft.nl/~risk/index.php?option=com_docman&task=doc_download&gid=190&&Itemid=13

[Ander04] C. Anderson, C. Kitts, A MATLAB Expert System for Ground-Based Satellite Operations, Proceedings of the 2005 IEEE Aerospace Conference, Big Sky, MT, USA, 2004. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1559682

[Beet05] M. Beetz, Probabilistic Hybrid Action Models for Predicting Concurrent Percept-driven Robot Behavior, Journal of Artificial Intelligence Research 24 (2005) 799-849 http://www.jair.org/media/1565/live-1565-2552-jair.pdf

[Bell04] D. Bellot, R. Siegwart, P. Bessiµere, A. Tapus, C. Coué, J. Diard, Bayesian Modeling and Reasoning for Real World Robotics: Basics and Examples, Lecture Notes in Computer Science (LNCS), Volume 3139/2004, Springer Berlin / Heidelberg, 2004 http://hal.archives-ouvertes.fr/docs/00/18/20/55/PDF/bellot04.pdf

[Benn00] B.H. Bennett, P. Bergstrom, G.D. Hadden, T. Samad, G.J. Vachtsevanos, J. Van Dyke, Application Challenges: System Health Management for Complex Systems, 5th International Workshop on Embedded HPC Systems and Applications (EHPC’2000), 2000
http://ipdps.cc.gatech.edu/2000/ehpc/18000785.pdf

[Brig08] C. Brignone, C. De Ambrosi, M. De Luca, F. Narteni, A. Tacchella, S. Verstichel, G. Villa, Engineering knowledge-based condition analyzers for on-board intelligent fault classification: a case study, 4th IET International Conference on Railway Condition Monitoring, RCM 2008
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04580850

[Byin03] C.S. Byington, P.W. Kalgren, R. Johns, R.J. Beers, Embedded Diagnostic/Prognostic Reasoning and Information Continuity for Improved Avionics Maintenance, Proceedings of AUTOTESTCON 2003, IEEE Systems Readiness Technology Conference.
http://www.impact-tek.com/Resources/TechnicalPublicationPDFs/Aerospace/Impact_AAV_DiagProgReasoningforAvionicsMaintenance_AutoTestCon.pdf

[Chang93] Yu Lo C. Chang, L.C. Lande, Homg-Shing Lut, M.T. Wells, Bayesian Inference for Fault Diagnosis in Real-Time Distributed Systems, Proceedings of the Second Asian Test Symposium, 1993
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=398827&isnumber=9011

[Chin2006] Chin Sun, K. Nguyen, Long Vu, S.C. Bisland, Prognostic/Diagnostic Health Management System (PHM) for Fab Efficiency, Advanced Semiconductor Manufacturing Conference, 2006. ASMC 2006. The 17th Annual SEMI/IEEE
http://www.qwiksinc.com/presentation/phm_fab_efficiency02a.pdf

[Clan89] W.J. Clancey, The Knowledge Level Reinterpreted: Modeling How Systems Interact, Machine Learning Journal, Volume 4, Numbers 3-4 / December, 1989, Springer Netherlands
http://www.springerlink.com/content/r3j5l76435q5w230/fulltext.pdf

[Costa08] P.C.G. Costa, M. Ladeira, R.N.Carvalho, K.B. Laskey, L.L. Santos, S. Matsumoto, (2008) A First-Order Bayesian Tool for Probabilistic Ontologies, In Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS-21), Palo Alto: AAAI Press, 2008
http://volgenau.gmu.edu/~klaskey/papers/2008_FLAIRS_CostaEtAl.pdf

[FSAP] M. Bozzano, A. Villafiorita, The FSAP/NuSMV-SA Analysis Platform, International Journal on Software Tools for Technology Transfer (STTT), Volume 9, Issue 1, Springer-Verlag, February 2007
http://sra.itc.it/tools/FSAP/dissemination/papers/sttt06.pdf

[Good89] R. M. Goodman, J. Miller, P. Smyth, H. Latin, Real Time Autonomous Expert Systems in Network Management, Journal: Information Technology & People, Vol. 5, Issue 3, Page 205 – 228, 1989
http://www.emeraldinsight.com/Insight/viewPDF.jsp?contentType=Article&Filename=html/Output/Published/EmeraldFullTextArticle/Pdf/1610050304.pdf

[Guo02] H. Guo, W. Hsu, A Survey of Algorithms for Real-Time Bayesian Network Inference, In the joint AAAI-02/KDD-02/UAI-02 workshop on Real-Time Decision Support and Diagnosis Systems, 2002
http://www.kddresearch.org/Workshops/RTDSDS-2002/papers/RTDSDS2002-GH-01.pdf

[Hopf93] H. Hopfmüller, W. Kehl, G. Schapeler, Application prototype of a model-based expert system to the maintenance of a MAN, Proceedings ISPIE Vol. 1975, 14, Local and Metropolitan Area Networks, 1993
http://spiedl.aip.org/getpdf/servlet/GetPDFServlet?filetype=pdf&id=PSISDG001975000001000014000001&idtype=cvips&prog=normal

[James01] M.L. James, An Autonomous Diagnostic and Prognostic Monitoring System for NASA's Deep Space Network, IEEE Aerospace Conference, Big Sky, Montana, USA, 2001
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=878248&isnumber=18972

[Jian03] L. Jianhui, M. Namburu, K. Pattipati, O. Liu, M. Kawamoto, S. Chigusa, Model-based prognostic techniques, Proceedings of the AUTOTESTCON 2003, IEEE Systems Readiness Technology Conference.
http://www.teamqsi.com/doc/063Luo.pdf

[Jun06] G. Junwen ,Q. Zheng, H. Xingchen, Research on Modeling the Nonhomogenous Markov Decision Systems with Dynamic Bayesian Network, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.6, June 2006
http://paper.ijcsns.org/07_book/200606/200606B04.pdf

[Kauf05] M.A. Kaufman, J.W. Sheppard, Bayesian modeling: an amendment to the AI-ESTATE standard, AUTOTESTCON 2005 Conference Record, IEEE Press, Sept. 2005
http://www.cs.jhu.edu/~jsquad/pubs/auto-2005b.pdf

[Lang06] C.J. Langmead, S.K. Jha, E. M. Clarke, Temporal Logics as query languages for Dynamic Bayesian Networks: Application to *D.Melanogaster* Embryo Development, Carnegie Mellon University School of Computer Science Technical Report CMU-CS-06-159, September 2006
http://reports-archive.adm.cs.cmu.edu/anon/2006/CMU-CS-06-159.pdf

[Lask06] K.B. Laskey, First-Order Bayesian Logic, George Mason University, VA, USA, 2006 http://volgenau.gmu.edu/~klaskey/papers/Laskey_FOBL.pdf

[Lee04] R. Lee, R. Watson, C. Kitts, P. Stang, B. Palmintier, Anomaly Detection Using the Emerald Nanosatellite On-Board Expert System, Proceedings of the 2004 IEEE Aerospace Conference, Big Sky MT, USA, 2004. http://hubbard.engr.scu.edu/docs/pubs/2004/Emerald%20Expert%20System.pdf

[Pai01] G.J. Pai, J.B. Dugan, Enhancing Software Reliability Estimation Using Bayesian Networks and Fault Trees, Proceedings of the IEEE International Symposium on Software Reliability Engineering (Fast Abstract Track), 2001 http://www.chillarege.com/fastabstracts/issre2001/2001113.pdf

[Patt] K. Pattada, On-Board, Real-Time Prognostic Health Management of Aircraft Engine Drive Train and Thrust Vectoring System, Impact Technologies, LLC http://www.virtualacquisitionshowcase.com/docs/2006/Impact2-Brief.pdf

[Port07] L. Portinale, A. Bobbio, D.C. Raiteri, S. Montani, Compiling Dyanamic Fault Trees into Dynamic Bayesian Nets for Reliability Analysis: the RADYBAN Tool, Proceedings of the Fifth UAI Bayesian Modeling Applications Workshop (UAI-AW 2007) http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-268/paper6.pdf

[RAS04] Editorial, Knowledge engineering and ontologies for autonomous systems: 2004 AAAI Spring Symposium, Robotics and Autonomous Systems 49, Elsevier, 2004 http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V16-4DJ4FD8-1&_user=6154828&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000063060&_version=1&_urlVersion=0&_userid=6154828&md5=b08dbe978c43b82c7e8749cc4d484b1a

[RAS08] Editorial, Using semantic knowledge in robotics, Robotics and Autonomous Systems 56 (2008) 875-877 http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V16-4T72WWW-1&_user=6154828&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=972973814&_rerunOrigin=google&_acct=C000063060&_version=1&_urlVersion=0&_userid=6154828&md5=f5439cf36a71644c02d1053f71e4a749

[RESS03] Guest Editorial, Safety, reliability and security of industrial computer systems, In: Reliability Engineering and System Safety, Volume 81, 235 – 238, 2003 http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6V4T-4938NB9-3-1&_cdi=5767&_user=6154828&_orig=search&_coverDate=09%2F30%2F2

003&_sk=999189996&view=c&wchp=dGLbVtb-zSkWb&md5=ba588f5dae050f45821e4d0a2d49d751&ie=/sdarticle.pdf

[Speel01] P-H. Speel, A.Th. Schreiber, W. van Joolingen, G. van Heijst 3, G.J. Beijer, Conceptual Modelling for Knowledge-Based Systems, Encyclopedia of Computer Science and Technology, Marce Dekker Inc., New York, 2001
http://www.cs.vu.nl/~guus/papers/Speel01a.pdf

[Stein03] M. Steinder, A.S. Sethi, Application of Bayesian Reasoning Techniques to Fault Localization in FCS Networks, Proc. CTA Annual Conference, College Park, MD, USA, 2003
http://www.cis.udel.edu/~sethi/papers/03/cta03-fault.pdf

[Wong96] L. Wong, F. Kronberg, a. Hopkins, F. Machi, P. Eastham, Development and Deployment of a Rule-Based Expert System for Autonomous Satellite Monitoring, Astronomical Data Analysis Software and Systems V, ASP Conference Series, Vol. 101, 1996
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.2019

[Wood04] R.T. Wood, J.S. Neal, C.R. Brittain, J.A. Mullens, Autonomous Control Capabilities for Space Reactor Power Systems, SPACE TECHNOLOGY AND APPLICATIONS INTERNAT. FORUM-STAIF 2004: Conf. on Thermophys. in Microgravity; Commercial/Civil Next Gen. Space Transp.; 21st Symp. Space Nuclear Power & Propulsion; Human Space Explor.; Space Colonization; New Frontiers & Future Concepts, AIP Conference Proceedings, Volume 699, pp. 631-638 (2004).
http://www.ornl.gov/~webworks/cppr/y2001/pres/119492.pdf

1.2.3 Bibliography

[Baier07] J.A. Baier, Procedural Domain Control Knowledge and State-of-the-Art Planners, Extended abstract of the ICAPS-07 paper by Baier, Fritz, & McIlraith, 2007
http://abotea.rsise.anu.edu.au/satellite-events-icaps07/dc/dc-03.pdf

[Bash03] H. Basher, J.C. Neal, AUTONOMOUS CONTROL OF NUCLEAR POWER PLANTS, Report ORNL/TM-2003/252 by Oak Ridge National Laboratory, Tennessee, USA for the U.S. Department of Energy, 2003
http://www.ornl.gov/~webworks/cppr/y2001/rpt/118800.pdf

[Bobb03] A. Bobbio, M. Gribaudo, A. Horvàth, Modeling a car safety controller using Fluid Stochastic Petri Nets, In: Proceedings 6-th International Workshop on Performability Modeling of Computer and Communication Systems (PMCCS6), pp 27-30, September, 2003.
http://web.unipmn.it/~bobbio/BIBLIO/PAPERS/ANNO03/pmccs6-2003.pdf

[Clan83] W.J. Clancey, The Advantages of Abstract Control Knowledge in Expert System Design, Stanford University, USA, Proceedings of the National Conference on Artificial Intelligence, 1983
http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/34/6e/ec.pdf

[Doria93] R. Doraiswami, M. Stevenson, C.P. Diduch, Autonomous Control Systems: Monitoring, Diagnosis, and Tuning, University of New Brunswick, Canada, 1993.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=384986&isnumber=8734

[Gabal03] A. Gabaldon, Compiling Control Knowledge into Preconditions for Planning in the Situation Calculus, In Proc. of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03
http://dli.iiit.ac.in/ijcai/IJCAI-2003/PDF/152.pdf

[Grib03] M. Gribaudoa, A. Horváth, A. Bobbiob, E. Troncic, E. Ciancamerlad, M. Minichino, Fluid Petri Nets and hybrid model-checking: a comparative case study, In: Reliability Engineering and System Safety, Volume 81, 239 – 257, 2003
http://www.sciencedirect.com/science?_ob=MiamiImageURL&_imagekey=B6V4T-4938NB9-4-5P&_cdi=5767&_user=6154828&_check=y&_orig=search&_coverDate=09%2F30%2F2003&view=c&wchp=dGLbVtb-zSkWb&md5=2ef2a25317ac07d51f9ed00316ea6220&ie=/sdarticle.pdf

[Jen88] P.G. Jenkins, Towards on-vehicle real-time knowledge based systems, IEEE Colloquium on Applications of Expert Systems in Road Transportation, 1988
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00208642

[Laur05] K.K. Laursen, M.F. Pedersen, J.D. Bendtsen, L. Alminde, The SOPHY framework: simulation, observation and planning in hybrid systems, Fifth International Conference on Hybrid Intelligent Systems, 2005. HIS '05.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1587789

[Li08] H.X. Li, B.C.Williams, Generative Planning for Hybrid Systems Based on Flow Tubes, Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008
http://groups.csail.mit.edu/mers/papers/Li-Williams-ICAPS08.pdf

[Lohm] G. E. Lohman, M. Briere, Autonomous Control of UAVs Today Through Behavioral Modeling, AMEWAS, Inc.
http://www.sisostds.org/index.php?tg=fileman&idx=get&id=2&gr=Y&path=CGF-BRIMS%2F13th+CGF-BR%2F13th+CGF-BR+Poster+or+Demo+Abstracts&file=04-BRIMS-095.pdf

[Lydia] The Lydia Project
http://www.st.ewi.tudelft.nl/~gemund/Lydia/index.html

[Miligh07] G. Milighetti, H.-B. Kuntze, Fuzzy based decision making for the discrete-continuous control of humanoid robots, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 29 - Nov 2, 2007, San Diego, CA, USA
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04399054

[PDK] Pdk/PDDL-K Reference Manual
http://pdk.dia.uniroma3.it/doc/userManual.ps

[Smets94] P. Smets, What is Dempster-Shafer's model?, Advances in the Dempster-Shafer theory of evidence, Pages: 5 – 34, John Wiley & Sons, Inc., New York, NY, USA, 1994
http://iridia.ulb.ac.be/~psmets/WhatIsDS.pdf

[Youn03] H.L.S. Younes, Extending PDDL to Model Stochastic Decision Processes, In Proceedings of the International Conference on Automated Planning and Scheduling Workshop on PDDL, 2003
http://www.tempastic.org/talks/icapsws2003b.pdf

[Zeig93] B. P. Zeigler, J. Kim, Extending The Devs-Scheme Knowledge-Based Simulation Environment For Real-Time Event-Based Control, IEEE Trans. on Robotics and Automation, 1993
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.9165&rep=rep1&type=pdf

1.3 Acronyms and abbreviations

The following acronyms are used or are relevant in this document:

Acronyms	Description
AADL	Architecture Analysis and Design Language
AR	Acceptance Review
BBN	Bayesian Belief Network
CDR	Critical Design Review
COTS	Commercial-Off-The-Shelf
CPT	Conditional Probability Table
CTMC	Continuous Time Markov Chain
DBN	Dynamic Bayesian Network
DDF	Design Definition File
DFT	Dynamic Fault Tree
DJF	Design Justification File

Acronyms	Description
DRD	Document Requirement Description
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
ESTEC	European Space research and Technology Centre
FDIR	Fault Detection Identification and Recovery
FOBL	First-Order Bayesian Logic
FSPN	Fluid Stochastic Petri Net
FT	Fault Tree
FTA	Fault Tree Analysis
GSPN	Generalised Stochastic Petri Net
IPR	Intellectual Property Rights
MDP	Markov Decision Process
MF	Maintenance File
MGT	Management File
MMI	Man-Machine Interface
MOTS	Modified-Off-The-Shelf
MRM	Markov Reward Model
OP	Operational Documentation
PAF	Product Assurance File
PDR	Preliminary Design Review
PN	Petri Net
QR	Qualification Review
RAMS	Reliability, Availability, Maintainability and Safety
RB	Requirement Baseline
SDP	Software Development Plan
SLC	Software Life Cycle
SPN	Stochastic Petri Net
SRR	System Requirement Review
SUM	Software User Manual
SVTS	Software Validation Testing Specification
TS	Technical Specification
UML	Unified Modelling Language

2 Background and Objectives

2.1 Background

Autonomous spacecraft operation relies on the adequate and timely reaction of the system to changes in the operational environment, as well as in the operational status of the system. The system environment can exhibit various degrees of non-deterministic behaviour, ranging from low to medium level for orbiting planetary observation spacecraft, to highly non-deterministic for planetary surface exploration cases.

The operational status of the system is dependent on the internal system dependability factors (e.g. sub-system and component reliability models), external environment factors affecting the system reliability and safety (e.g. thermal, radiation, illumination conditions), and system-environment interaction profiles directly affecting system dependability and safety (e.g. stress factors, resource utilisation profiles). Combinations of these factors cause mission execution anomalies, including mission degradations and system failures. To address possible system faults and failures, mitigation means are built into the system in the form of Health Management System, for the biggest part relying on the Fault Detection, Isolation and Recovery (FDIR).

Currently employed FDIR operation is based on the design-time analysis of the faults and failure scenarios (e.g. FMEA, FTA) and run-time observation of the system operational status (health monitoring). It relies on predefined countermeasures to prevent faults from causing the system failures (if possible), and has the main objectives of: (1) timely detection of faults, (2) initiation of the corresponding predefined recovery actions. If no corresponding action could be found, FDIR proceeds by executing the safing actions to put the spacecraft into a known safe configuration and transfers control to the Ground operations for troubleshooting and planning the recovery actions.

This approach however, suffers from multiple shortcomings, which, depending on the operational context, may significantly reduce effectiveness or undermine adequacy of the FDIR procedures, for example:

- The system, as well as its environment, is only partially observable by the FDIR monitoring. This introduces uncertainty in the interpretation of observations.
- The FDIR represents a reactive approach, a post-factum operation, not capable of preventive measures. It cannot provide and utilise prognosis for the imminent failures.
- Cases of deferring the decisions to the Ground operations originate from the need of the knowledge and experience of Ground operators to analyse and evaluate the System and environment status in order to come up with specific course of recovery actions. As this kind of system-wide causal knowledge is not present on-board, automated FDIR procedures cannot leverage it.

- Knowledge of the general operational capabilities of the System is not represented on-board. This makes it impossible to estimate the impact of the occurred faults and failures on these capabilities. A Ground intervention is thus required for evaluation and prognosis for the already planned operations and upcoming operations planning. Without this causal knowledge and probabilistic success criteria the on-board plan execution may spend resources on the execution of the current plan, which is to fail in its future steps due to the compound effect of the local faults/degradations on the global system capabilities.

Multiple studies have addressed the issues of prognosis and knowledge-based reasoning, some of them for the on-board systems.

Prognostic reasoning is based on discovery and encoding of the dependencies and correlations between the set of observation parameters and the resulting estimations of interest. They are based on probabilistic techniques and various forms of causal modelling, capturing reasoning rules based on system knowledge and expert judgement. While capable of providing adequate results for the statically captured system configurations, they can poorly deal with the dynamic aspects of the systems, such as recovery actions and reconfiguration.

Approaches to the system failure scenario analyses, such as FMEA and FTA, capture mostly static system configurations, without taking into account dynamic and temporal aspects of the system and do not address evolution of the system characteristics and history of the system interaction with the environment.

The FDIR procedures do not reflect probabilistic causal dependencies between the faults and general system capabilities on one hand, and between the system-environment interaction evolution and system dependability characteristics on the other hand.

To address these issues an approach to on-board FDIR is needed which has the capability to reason about anomalous observations based on the global knowledge of the system and its capabilities, system environment, and system-environment interaction in the presence of uncertainty. It has to provide the system with prognosis on the operational status to be taken into account for autonomous operational planning and to allow preventive recovery actions.

This approach will build upon existing research on automated probabilistic dependability and safety analysis, causal probabilistic modelling and analysis under uncertainty, and knowledge-based techniques. It will envisage integration of the techniques from currently different application domains to allow autonomous on-board FDIR reasoning under uncertainty and partial observability, utilising system and environmental knowledge, and inferring prognosis on system operational capabilities.

2.2 Objectives of the activity

This study will develop a demonstrator and perform proof-of-concept case studies for the innovative FDIR element of an autonomous spacecraft. This will enable on-board real-time reasoning, analyses, and estimations for Prognostic Health Management, Dependability and

Safety, and inference of anomaly resolution approaches. It will provide for reasoning about the impact of system and environment state on spacecraft capabilities and mission execution to be factored in the autonomous planning and re-planning.

The objectives of the activity are:

- to increase mission return,
- to enable the design of more capable future missions,
- to reduce the cost of Ground operations, and
- minimise the space segment assets downtime due to anomalies.

Space systems must be capable of on-board decision making to appropriately react to system faults and failures, proper estimation of failure impacts on system operation, and adequate resolution of the encountered anomalies.

To achieve this goal, a knowledge of the system capabilities, its operational environment, and knowledge of anomaly resolution strategies, have to be represented on-board and made available for autonomous reasoning. To allow for such reasoning these aspects of system representation require a common modelling framework, encompassing timed stochastic behavioural and fault modelling, representation of the hybrid aspects of the system (e.g. reasoning on continuous resources), stochastic causal modelling, knowledge-based modelling.

The global objective of this study is to demonstrate that integration of the innovative technologies (i.e. model-based autonomy, model checking of stochastic hybrid models, run-time Dependability and Safety analysis, causal modelling, probabilistic calculus, Knowledge-Based Systems) in a unified modelling and autonomous reasoning framework may increase the achievable level of autonomy. The main focus is on the autonomous anomaly resolution and prognostic pro-active FDIR capabilities.

This global objective comprises the following sub-objectives:

1. Evaluation and justification of an integrated and unified use of the stochastic hybrid model checking, causal probabilistic techniques and Knowledge-Based approaches, suited for on-board automated analysis, to increase the space systems level of autonomy in terms of anomaly resilience and autonomous recoverability;
2. Definition of an integrated modelling framework for specification of the models suited for on-board autonomous reasoning to infer system Health, Dependability and Safety status and prognosis, and (preventive) anomaly resolution approaches;
3. Development of an on-board software prototype, the *Anomaly Resolution and Prognostic Health management for Autonomy* (ARPHA), implementing the required autonomous reasoning and inference techniques, based on the use of stochastic hybrid model checking and probabilistic calculus approaches;
4. Demonstration of the approach on case studies involving autonomous on-board systems and evaluation of the experimental results in terms of applicability, scalability, and performance;

5. Evaluation of adequacy of the approach and developed technology for use in the context of critical on-board space systems.

3 Work to be performed

3.1 Work Logic

The work to be carried out in this project shall comprise the following tasks:

1. Production of the Requirements Baseline (RB)
2. Production of the Technical Specification (TS)
3. Design, Coding, Verification and Validation against the TS and RB
4. Performance Evaluation and Characterisation of the Approach

This decomposition of tasks reflects the Software Development Process as described in the ECSS Software Standard [E-40C]. Table 1 presents the study parts and their constituent tasks.

Part 1	***Production of the Requirements Baseline*** TASK 1.1: Synthesis on Anomaly Resolution, Failure Impact Analysis and Prognostic FDIR needs and Potential Solutions TASK 1.2: Requirements Baseline Elicitation
Part 2	***Production of the Technical Specification*** TASK 2.1: Software Specification Elicitation TASK 2.2: Architectural Design
Part 3	***Implementation of the Anomaly Resolution and Prognostic Health management for Autonomy (ARPHA)*** TASK 3.1: Detailed Design, Coding, and Validation against the TS TASK 3.2: Validation against the RB
Part 4	***Performance Evaluation*** TASK 4.1: Evaluation of the Approach on a Case Study TASK 4.2: Characterisation of the Approach

Table 1: Work Logic summary

The tasks are further detailed in the following sections.

The following major reviews shall be held:

- System Requirements Review (SRR), in conclusion of TASK 1.2
  - This review will evaluate the proposed approach and technologies, adequacy of the Requirements Baseline, and suitability of the proposed case studies.
- Preliminary Design Review (PDR), in conclusion of TASK 2.2
  - This review will evaluate adequacy of the software Technical Specification and Architectural Design with regard to the proposed solution and RB.
- Critical Design Review (CDR), in conclusion of TASK 3.1
  - This review will evaluate adequacy and completeness of software design with regard to TS, will evaluate software Validation activities with respect to the TS, and will analyse Design Justification.

- Acceptance Review (AR), in conclusion of TASK 4.2
    - This review will evaluate the overall achievement of the study objectives. It will analyse adequacy, effectiveness and applicability of the study results in the context of the space on-board software systems.

3.2 Production of the Requirements Baseline

3.2.1 Task 1.1: Synthesis on Anomaly Resolution, Failure Impact Analysis and Prognostic FDIR needs and Potential Solutions

- Input
    - This Statement of Work
    - Reference Documents (Section 1.2.2)
- Task description:
    - This Task shall provide a detailed overview of the state-of-the-art with regard to the Model-Based and Knowledge-Based automated reasoning in the context of autonomous Failure and Anomaly Resolution, Failure Impact Analysis, and System Health Prognosis. It shall also address autonomous System Dependability and Safety evaluation.
    - It shall concentrate on the use of these approaches for the autonomous spacecraft operation and will evaluate various modelling formalisms suited for representation of the dynamic failure and recovery scenarios, causal probabilistic relations between the operational/environmental conditions and fault occurrences, causal probabilistic relations between the failures/anomalies and operational system capabilities, and propose a unifying modelling framework.
    - Corresponding formal and probabilistic analysis techniques suited for run-time on-board use will be evaluated.
    - This Task will propose a set of technologies, based on combination of model checking for stochastic hybrid systems, causal probabilistic inference, and Knowledge-Based inference techniques. The modelling framework and the automated analysis techniques shall form the basis for the definition of the on-board *Anomaly Resolution and Prognostic Health management for Autonomy* (ARPHA) software Building Block.
    - This Task shall survey and propose the case studies suitable and appropriate for the practical evaluation of the proposed approach.
- Output / Approval conditions
    - Technical Note detailing the state-of-the-art in the Model-Based and Knowledge-Based automated reasoning in the context of autonomous

Failure and Anomaly Resolution, Failure Impact Analysis, and System Health Prognosis, as well as autonomous System Dependability and Safety evaluation; proposed approaches and technologies as basis of the project; selected case studies.

3.2.2 Task 1.2: Requirements Baseline Elicitation

- Input
  - This Statement of Work
  - Reference Documents (Section 1.2.2)
  - Output of the Task 1.1
- Task description:
  - This Task shall be dedicated to the specification of the System Requirements related to the software for the Anomaly Resolution and Prognostic Health management for Autonomy (ARPHA) Building Block. It shall provide an overview of the system to be developed and a draft UML architectural model of the ARPHA shall be produced reflecting the logical structure and high-level functional analysis.
  - The ARPHA architectural model shall be put in the context of autonomy operation and spacecraft avionics. Requirements for the spacecraft avionics and (autonomous) software architecture in line with the ARPHA objectives shall be elucidated.
  - This Task shall be concluded with the SRR.
- Output / Approval conditions
  - The Requirements Baseline
  - The draft avionics model (AADL/SysML/UML), draft global autonomous software architectural (context) model (UML), ARPHA draft architectural UML model

3.3 Production of the Technical Specification

3.3.1 Task 2.1: Software Specification Elicitation

- Input
  - This Statement of Work
  - Reference Documents (Section 1.2.2)

  - Outputs of the Task 1.1 and Task 1.2
- Task description: This Task shall produce the Software Specification based on the Requirements Baseline. It will detail architectural and design choices, as well as selection of technologies and software components (to be developed or (re-)used).
- Output / Approval conditions
  - Software Specification Document

3.3.2 Task 2.2: Architectural Design

- Input
  - This Statement of Work
  - Reference Documents (Section 1.2.2)
  - Output of the Task 2.1
- Task description: This Task shall produce a detailed UML-based architectural design of the ARPHA. It shall reflect upon interfacing with the ARPHA operational context elements of the avionics and (autonomous) spacecraft software. This Task shall be concluded with the PDR.
- Output / Approval conditions
  - Architecture Design Document

3.4 Implementation of the Anomaly Resolution and Prognostic Health management for Autonomy (ARPHA)

3.4.1 Task 3.1: Detailed Design, Coding, and Verification against the TS

- Input
  - This Statement of Work
  - Reference Documents (Section 1.2.2)
  - Outputs of the Task 2.1 and Task 2.2
- Task description: This Task shall perform full development of the ARPHA. This includes detailed design, coding, unit testing, preliminary integration testing, and validation against the Software Specification. The design model shall be expressed in UML and represent a refinement of the Architectural Design. This Task shall be concluded with the CDR
- Output / Approval conditions

  - Detailed Design
  - Implementation code
  - TS-Validation Test Report

3.4.2 Task 3.2: Validation against the RB

- Input
  - This Statement of Work
  - Reference Documents (Section 1.2.2)
  - Outputs of the Task 2.1, Task 2.2 and Task 3.1
- Task description: This Task shall perform the ARPHA validation against the Requirements Baseline.
- Output / Approval conditions
  - ARPHA software building block final product
  - RB-Validation Test Report
  - Software documentation, including the User Manual

3.5 Performance Evaluation

3.5.1 Task 4.1: Evaluation of the Approach on a Case Study

- Input
  - This Statement of Work
  - Reference Documents (Section 1.2.2)
  - Outputs of the Task 1.1, Task 2.1, Task 2.2, Task 3.1 and Task 3.2
- Task description: This Task shall perform empirical evaluation of the approach on case studies. The case studies shall be representative of the autonomous spacecraft operation.
- Output / Approval conditions
  - Technical Report representing the results of the evaluation.

3.5.2 Task 4.2: Characterisation of the Approach

- Input
  - This Statement of Work

- Reference Documents (Section 1.2.2)
- Outputs of the Task 1.1, Task 2.1, Task 2.2, Task 3.1, Task 3.2 and Task 4.1

- Task description: This Task shall provide the characterisation of the overall approach in terms of adequacy, effectiveness, reliability, availability, and performance. It shall also conclude on the adequate general software and avionics architecture to support the ARPHA functionalities. This Task shall be concluded with the AR.
- Output / Approval conditions
    - Technical Report presenting the characterisation of the global approach and the adopted software and avionics architecture.
    - Final Report

4 Requirements for Management, Reporting, Meetings and Deliverables

The standard requirements for Management, Reporting, Meetings and Deliverables (Appendix 2 to the Contract) shall apply, taking account of the following specific requirements for the present activity, which shall prevail in case of conflict.

4.1 Management

Section 1 of the standard requirements shall apply.

4.2 Reporting

Section 2 of the standard requirements shall apply.
The Contractor shall provide a monthly progress report covering the work executed. This report shall give a description of progress, reasons for potential slippages and corrective actions, work planned for the next reporting period, expected dates for the major schedule items, and an updated schedule.

The Contractor shall notify the Agency’s representatives (Technical Officer and Contracts Officer) of any event likely to cause major delays to the time schedule of the work programme or significantly impact the scope of the work to be performed.

As soon as they become available and always within the time frame agreed by the Agency, the Contractor shall submit for the Agency’s approval all technical notes, which are produced during the execution of the Contract. Any technical documentation to be discussed at a meeting with the Agency shall be submitted at least two weeks prior to such a meeting.

Note that all documents mentioned shall also be delivered as an electronic file.

4.3 Meetings

Section 3 of the standard requirements shall apply.
The Agency intends to monitor the execution of the Contract through dedicated meetings: the Kick-Off Meeting, Progress Meetings, and a Final Presentation.

The Kick-Off meeting has to be considered as the first event in the project. It will occur after the negotiation meeting to approve formally the technical baseline.

Progress Meetings will take place at a frequency to be determined by the Agency. That frequency could be changed if difficulties occur during the Contract execution requiring further discussions. It shall be possible to arrange progress meetings at the Contractor’s premises when/if required.

The objective of each meeting is to assess the results of the technical efforts for completeness, correctness and compliance with the requirements, and to verify the achievement of the objectives. Shortcomings, problems, corrective actions, and potential changes will be identified and formally addressed.

A Final Presentation will be scheduled after formal approval of all deliverables. The Contractor shall make a presentation summarising all the main activities and achievements made during the Contract execution. All the deliverable items shall be available to ESA within the date of the final presentation.

Additional meetings are not excluded, and either the Agency or the Contractor may request ad hoc meetings.

The Contractor is responsible for the preparation and distribution of minutes of all meetings held in connection with the Contract. The minutes shall clearly identify all agreements made and actions accepted at the meeting together with an update of the action item list and the document list. A draft shall be signed at the end of every meeting.

4.4 Deliverables

4.4.1 Documentation

Document identifier	Title	Milestone	Number of copies
D1	Technical Note: Model-Based and Knowledge-Based reasoning for Anomaly Resolution and Prognostic Health Management.	End of Task 1.1	1
D2	Requirements Baseline	End of Task 1.2	1
D3	Software Specification Document	End of Task 2.1	1
D4	Architecture Design Document	End of Task 2.2	1
D5	Detailed Design	End of Task 3.1	1
D6	Test Report: Software Validation with respect to the TS	End of Task 3.1	1
D7	Test Report: Software Validation with respect to the RB	End of Task 3.2	1

D8	ARPHA Software Documentation	End of Task 3.2	5
D9	ARPHA User Manual	End of Task 3.2	5
D10	Technical Report: Case studies evaluation	End of Task 4.1	1
D11	Technical Report: Characterisation of the approach	End of Task 4.2	1
D12	Final Report	End of Task 4.2	1
D13	Summary Report	End of Project	5 CD-ROMs +Paper copy
D14	Technical Data Package	End of Project	5 CD-ROMs +Paper copy
D15	User Manuals	End of Project	5 CD-ROMs +Paper copy

4.4.2 Hardware

Any hardware produced or procured under the contract shall be delivered to the Agency.

4.4.3 Software

Item identifier	**Description**	**Milestone**	**Number of copies**
SW1	ARPHA software (object code and source code)	End of Task 3.2	1

Any software produced or procured under the contract shall be delivered to the Agency.

4.5 Commercial Evaluation

A report in Commercial Evaluation according to Section 5 of Appendix 2 to the Contract is not required.

5 Schedule and Milestones

5.1 Duration

The duration of the work shall not exceed 18 months from kick-off to end of the activity (delivery of final report or hardware or software).

5.2 Milestones

The following milestones shall apply:

- MIL1: End of Part 1 of the project (see Table 1) concluded with the SRR and formal approval of the corresponding Tasks;
- MIL2: End of Part 2 of the project (see Table 1) concluded with the PDR and formal approval of the corresponding Tasks;
- MIL3: End of Task 3.1 of the project (see Table 1) concluded with the Prototype CDR and formal approval of the corresponding Task;
- MIL4: End of Part 4 of the project (see Table 1) concluded with the AR and formal approval of the corresponding Tasks;

5.3 Reviews

The following reviews shall be held:

5.3.1 System Requirements Review

- Date: Conclusion of TASK 1.2
- Location: Contractor’s premises
- Input: D1, D2, Draft D4
- Description: This review will evaluate the proposed approach and technologies, adequacy of the Requirements Baseline, and suitability of the proposed case studies.
- Output: SRR Report, prepared by the Contractor

5.3.2 Preliminary Design Review

- Date: Conclusion of TASK 2.2
- Location: Contractor's premises
- Input: D3, D4
- Description: This review will evaluate adequacy of the software Technical Specification and Architectural Design with regard to the proposed solution and RB.
- Output: PDR Report, prepared by the Contractor

5.3.3 Critical Design Review

- Date: Conclusion of TASK 3.1
- Location: Contractor's premises
- Input: D5, D6
- Description: This review will evaluate adequacy and completeness of software design with regard to TS, will evaluate software Validation activities with respect to the TS, and will analyse Design Justification.
- Output: CDR Report, prepared by the Contractor

5.3.4 Acceptance Review

- Date: Conclusion of TASK 4.2
- Location: ESA/ESTEC
- Input: D7, D8, D9, D10, D11, D12, SW1
- Description: This review will evaluate the overall achievement of the study objectives. It will analyse adequacy, effectiveness and applicability of the study results in the context of the space on-board software systems.
- Output: AR Report, prepared by the Contractor

6 Agency Undertakings

6.1 Customer Furnished Items

This activity is related to the previous ESA TRP project: On-Board Model Checking [OBMC], performed under the ESA Contract 20580/07/NL/JD. As this activity is considered logically extending/complementing the results of the OBMC project, the results of the latter, including the source code, can be provided to the Contractor, if necessary for the execution of this activity, for exclusive non-transferable use within the frame of execution of this activity, upon request.

Category	Description	Quantity	Delivery date
Documentation	OBMC Project Technical Data Package	1	Upon request
Software	OBMC Project software, including the source code, for the Autonomous Reasoning Engine (ARE) building block.	1	Upon request

6.2 Other Agency Undertakings

No other Agency undertakings are associated to execution of this activity.

Annex A: Technical Background

A.1 Background overview

This study takes as basis the established modelling and automated analysis approaches used in the respective domains of Safety and Dependability analysis, Fault Diagnosis, Prognostic Health Management, and Knowledge-Based applications (e.g. Decision Support Systems), with particular focus on the following:

- System Safety and Dependability analysis, based on Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA), and stochastic approaches such as Markov Chains and Stochastic Petri Nets;
- Fault Propagation evaluation, based on Model Checking of stochastic hybrid systems;
- Causal modelling, quantitative evaluation and reasoning under uncertainty, based on Bayesian Belief Networks and probabilistic calculus;
- Knowledge representation and Knowledge-Based reasoning, based on ontologies and probabilistic (Bayesian) inference approaches.

Fault Detection, Identification and Recovery (FDIR) systems employed on-board the spacecraft are based on the design-time safety and dependability analyses of the system. They are built to react to the available on-board diagnosis data and execute the a-priori defined recovery procedures.

In case no corresponding predefined procedure is available, the FDIR usually commands the spacecraft into the known (predefined) safe configuration, effectively transferring control for further diagnosis, analysis, and recovery actions to Ground.

For cases where timely resolution of the anomalous conditions is required, the conventional FDIR approach does not provide an adequate solution. Ground communication constraints (e.g. time delay, Ground visibility, limited bandwidth) and decision making time could jeopardise the spacecraft safety or mission success in the context of the dynamic spacecraft environment or spacecraft-environment interaction profile (e.g. in case of planetary rovers). Another factor affecting the adequacy of the Ground decisions is partial observability of the system and its environment, resulting in uncertainty of the analyses.

In cases where on-board FDIR does find available predefined recovery procedures, their success could be hampered by the uncertain/unforeseen context in which they are to be executed. This situation, opposite to the recovery objectives, could worsen the situation and increase the risk of further mission degradation or failure. The same consequences can also be caused by evolution of the original, design-time, estimations of the system (sub-systems) safety and dependability characteristics which are in reality affected by the mission execution history, environmental effects, and interaction with the environment [Clan89]. This leads to a situation where results of the recovery actions could not be reliably foreseen.

In the context of autonomous systems, where (re-)planning and scheduling is performed on-board, the likelihood of the plan execution success strongly depends on the reliability of assumptions used in the plan generation process. These assumptions comprise operational status of the system, state of the environment, and global operational system capabilities (in the context of environmental conditions and mission execution history). The latter are strongly dependent on the impact of occurred faults and failures and evolving safety and dependability characteristics. Generally this knowledge is only present on Ground, which increases the risk of autonomously producing plans with the low eventual probability of success lowering the mission efficiency and return, and waists the system resources.

Various lines of research are focusing on the discussed issues, however not in a coherent manner. The most widely used method of safety and dependability evaluation is application of FTA and FMEA which are performed during the system design. The Model-Based Development approaches allow to automatically generate the Fault Trees and FMEA tables from the Operational Models extended with the Fault Models [FSAP]. Annotating the faults with the occurrence probability enables simple quantitative evaluation of the probabilities of top-level events.

To account for the dynamical aspects of the system, including temporal aspects, recovery actions, and timing, the nominal Fault Tree formalism has been extended with the dynamic gates. While the Dynamic Fault Trees (DFT) are more representative, including additionally required elements, such as likelihood of recovery actions success, the analysis becomes significantly more complex. Such analyses can be performed through DFT translation to various flavours of Markov Chains (e.g. CTMC) or Markov Decision Processes with subsequent application of timed stochastic model checking approaches. If enabled to be used in real-time on-board, these techniques would broaden the FDIR autonomy capabilities. However, these approaches currently do not take into account the dynamic nature of the fault probabilities, which are causally related to the system operation history and cumulative environmental effects. These methods can be used for diagnosis purposes, but can not evaluate the failure impact on system operation as this requires broader knowledge of the system and its causal dependencies.

Causal probabilistic modelling and inference is applied in the domains of probabilistic estimation, especially where automated reasoning with incomplete and uncertain knowledge is required. The most widely used is Bayesian approach and use of the Bayesian Belief Networks (BBN). Use of probabilistic calculus allows for predictive (top-down), diagnostic (bottom-up), and mixed (trade-off-like scenario) inference on BBNs. These approaches find their application in reliability estimation [Pai01] and diagnostic systems [Stein03], [Chang93]. While mainly used off-line, run-time use of Bayesian approaches can significantly enhance the capabilities of autonomous systems by enabling reasoning under uncertainty, imperfect and incomplete observation data [Bell04]. Therefore, the BBN inference algorithms suited for the real-time and embedded use are needed [Guo02].

Another line of research is addressing a combination of system safety and dependability analyses along with the design information, expressed in the Fault Trees, with probabilistic assessment approaches, based on BBNs [Pai01]. This allows combining the FTA with the expert knowledge and probabilistic (causal) dependencies. Also approaches of complementary use of the Fault Trees and BBNs are being investigated, addressing the use of BBNs for estimating the probabilities of the basic faults in the Fault Trees [Ale07], allowing to incorporate the probabilistic environmental effects on system safety and dependability characteristics.

While providing good analysis tools, the Fault Tree formalism is based on binary logic and can not express probabilistic dependencies and non-binary multi-state elements. The BBNs are well suited for this purpose, in addition allowing for probabilistic logical operators and sequential dependencies [RESS03]. To leverage these abilities, a research effort is addressing (automated) translation of the Fault Trees into BBNs to integrate the analysis facilities of both [Ale07]. The resulting models shall allow for wider scope of analysis and more complex failure modes, including the sensitivity analysis on the sub-system and component contribution to the overall system characteristics.

However, as BBNs are essentially probabilistic Directed Acyclic Graphs (DAG), they fail to include any cyclic elements (returning the system into some previous state), such as possible recovery actions of the system, thus excluding the FDIR measures from the overall analysis [RESS03]. To alleviate this shortcoming and enable capturing dynamic aspects of the failure modes, the use of DFTs and their automated translation to the Dynamic Bayesian Networks (DBN), with corresponding analysis techniques, is being investigated [Port07]. This approach also brings the representation formalism in the domain of Markov Chains (CTMC) [Port07] and Markov Decision Systems [Jun06], enabling the application of existing analysis techniques, including the model checking of stochastic systems [Lang06].

Research on Bayesian approaches also extends to the domain of Knowledge-Based Systems. The main issues in this area are knowledge representation (formal encoding) and formal reasoning under uncertainty. To address these issues the First-Order Bayesian Logic (FOBL) has been introduced, which integrates the classical first-order logic with probability [Lask06]. It provides a logical foundation for knowledge representation. This approach is being further developed into the Knowledge-Based domain through introduction of the Bayesian Networks and FOBL techniques for representation of and formal reasoning on the Probabilistic Ontologies [Costa08].

Operation of an autonomous system, especially in the perspective of FDIR and Anomaly Resolution, reflects the execution of tasks which are, to large extent, nominally performed on Ground by operations and system experts. Resolution of non-trivial cases relies on expert judgement and experience, which represent accumulated domain knowledge. To allow for autonomous system to make use of this knowledge for its operation, this knowledge has to be available on-board in a form suited for automated (formal) reasoning. Moreover, the recovery/resolution approaches are operational context-dependent and can become inadequate in a pre-programmed form. While Knowledge-Based Systems (KBS) have a long history of research and application in Ground-based systems (e.g. Decision Support Systems, Diagnostic Systems, Health Monitoring) [Hopf93] [Good89] [Ander04] [Wong96], their on-board use for autonomous systems is attracting a renewed interest in the last years [RAS04]. Recent investigations in this area addressed the use of KBS in the context of on-board space and railway systems: satellite anomaly detection system [Lee04]; autonomous control of an on-board nuclear power system [Wood04]; intelligent locomotive fault classification system [Brig08].

The specifics of real-time autonomous systems introduce additional dimension of evolution of the system and its environment, and consequently the knowledge about them. This brings the need of reasoning with parametric, dynamic knowledge; combining and representing a-priory and in-situ knowledge; using the knowledge in the frame of model-based control [RAS04]; using specific modeling approaches for KBS [Speel01]. While various approaches exist for explicit knowledge representation for autonomous systems, the use of Bayesian Networks, reflecting the probabilistic approach [RAS08], appears to be most promising for reasoning under uncertainty and with incomplete knowledge. Bayesian approaches also allow for prognostic reasoning as they can capture causal probabilistic dependencies.

Prognostic techniques allow identifying imminent faults and failures, estimate the time to failure, and predict performance degradations. These techniques are being successfully applied in the domain of System Health Management and System Maintenance, were they increase the preventive maintenance efficiency and extend the operational life of systems.

Prognostic Health Management (PHM) systems often rely on the acquired expert knowledge and employ KBS techniques, mostly based on pattern/scenario recognition identifying the early stages of the imminent faulty conditions [Chin2006]. While acting upon prognostic estimations mainly takes the form of an off-line pro-active maintenance, the real-time data acquisition and prognostic analysis are starting to be used in large-scale applications [Benn00]. This enables automatic fault identification in the context of the compound effect of multiple faults, failure prediction, tracking the component and system degradation, maintenance scheduling, and corrective estimation of corrupted measurements. The PHM techniques can further reduce operational (and maintenance) costs if the inferred prognoses can be dealt with in an automated way based on the KBS approaches [James01].

Developments in this field show high potential for applying the BBN-based technologies in the field of PHM for real-time embedded avionics systems [Byin03] [Patt]. This approach allows for combining standard FDIR operation with operational context information and models of usage (history). It requires model-based prognostic techniques combining the design-level knowledge of the system with knowledge of the system's operational conditions and history [Jian03], allowing for the situated automata approach. To achieve this objective, prognostic analysis can be addressed through creating a temporally related set of BBNs, effectively constituting a DBN, capturing the prognostic relationships [Kauf05].

A.2 Envisaged solution

Previous ESA activity, the On-Board Model Checking [OBMC], has addressed the spacecraft autonomy and the use of model checking technology for its implementation. It has developed the Autonomous Reasoning Engine (ARE), providing the abstract model-based decision logic and procedures for the Deliberative and Executive Layers of autonomy, focusing on goal-driven planning and plan execution. The ARE has addressed the uncertainty aspects from the perspective of non-determinism of the commanding outcomes and the environment behaviour, and partial observability of the system status and its environment. The ARE includes an integrated FDIR module, which evaluates the assumed operational state of the system, incorporates recovery actions into produced planning, and validates executability of the plans.

While the OBMC activity has addressed the core functionalities of the goal-oriented autonomy, the proposed activity will consider the advanced FDIR approaches, which would allow for:

- Evaluation of failure impact on the operational capabilities of the system and on the currently executing plan;
- System status prognosis and its possible uses in autonomous planning and for pro-active autonomous measures to increase system Dependability and Safety;
- Inclusion of the environmental effects, system operation evolution, and system-environment interaction history into the diagnostic/prognostic reasoning in terms of causal probabilistic dependencies and available system and environment knowledge.

This activity will be concerned with the development of the Anomaly Resolution and Prognostic Health management for Autonomy (ARPHA) system. It will provide the model-based on-board reasoning using Knowledge-Based approaches, probabilistic calculus, and formal techniques (e.g. DBN inference, FOBL, CTMC analysis, model-checking of hybrid stochastic systems).

The system DFTs will represent a starting point for creation of the operational models in the basis of ARPHA reasoning. Knowledge about the environmental and system operational evolution effects on the parameters (e.g. probabilities) of the basic faults in the DFTs will be captured in a causal probabilistic Fault Estimation Model and represented in a form of BBN/DBN. Knowledge about the environment, system operational evolution, and system-environment interaction history will be captured in a causal probabilistic Failure Impact Estimation Model represented in a form of BBN/DBN, where a-priori failure probabilities will be estimated by the system DFT. Combination/integration of these three models will provide a coherent representation and estimation of system diagnoses/prognoses in the context of its environment and mission execution history.

A technique will be developed for a semantically correct translation of the combined/integrated model into a single formalism suited for a unified reasoning approach to be used by ARPHA. The DBN appears to be a good candidate for this unified formalism and further translation of the DBN model to a Markovian representation (e.g. CTMC, MDP, MRM) will be addressed. Representation most suitable for the effective real-time reasoning will be selected.

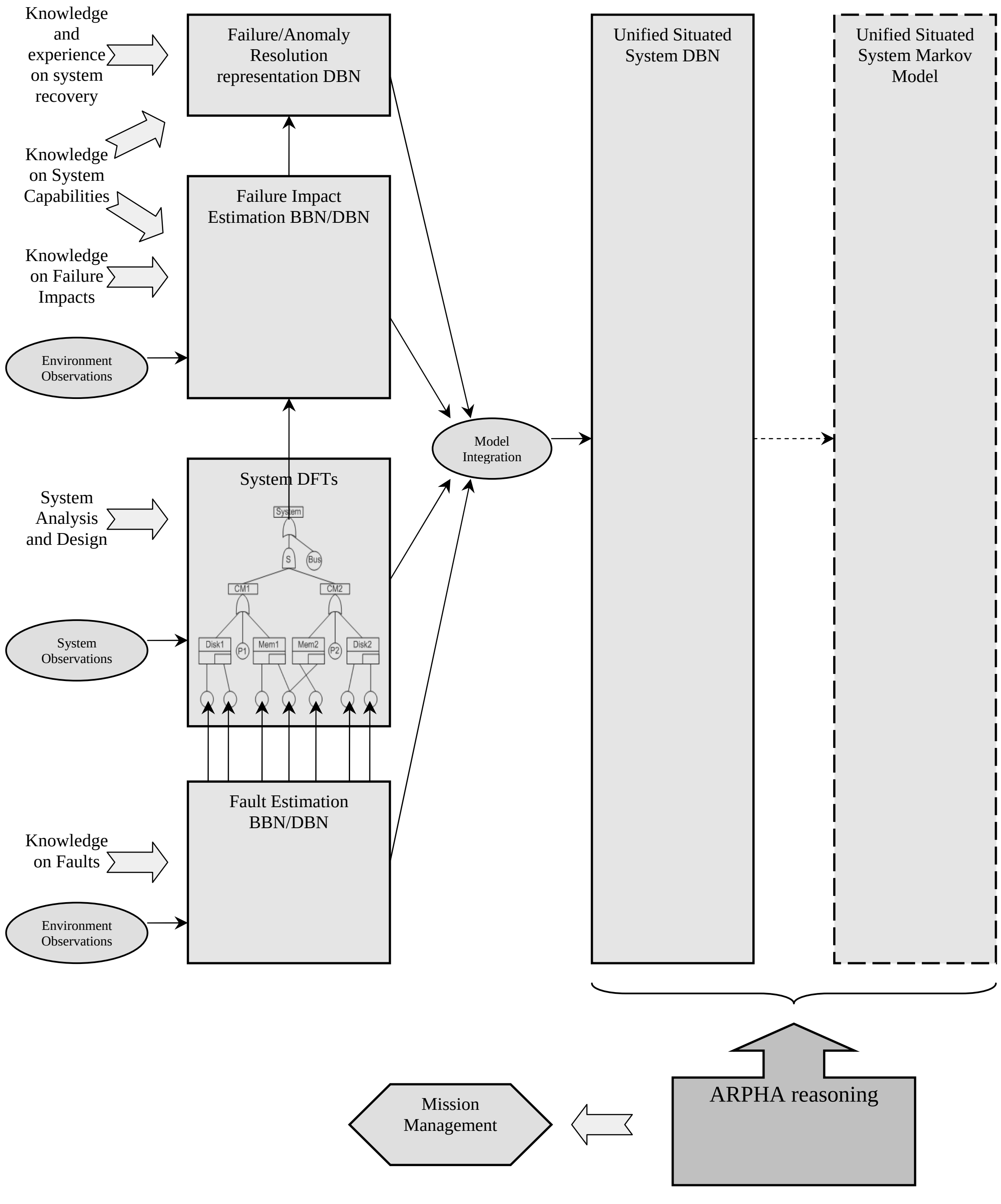
Knowledge and experience on system recovery
Failure/Anomaly Resolution representation DBN
Unified Situated System DBN
Unified Situated System Markov Model
Knowledge on System Capabilities
Failure Impact Estimation BBN/DBN
Knowledge on Failure Impacts
Environment Observations
Model Integration
System DFTs
System
S
Bus
CM1
CM2
Disk1
P1
Mem1
Mem2
P2
Disk2
System Analysis and Design
System Observations
Fault Estimation BBN/DBN
Knowledge on Faults
Environment Observations
ARPHA reasoning
Mission Management

Figure 1: Representation of an envisaged solution

Annex B: ECSS-E-ST-40C Tailoring

B.1

Documents may be merged upon request from the Contractor and approval by the Agency.

Requirement identification	Expected Output	Applicable
5.2.2.1a.	Specification of system requirements allocated to software	Y
5.2.2.1a.-a.	Functions and performance system requirements allocated to software	Y
5.2.2.1a.-b.	Verification and validation product requirements	Y
5.2.2.1a.-c.	Software operations requirements	Y
5.2.2.1a.-d.	Software maintenance requirements	N
5.2.2.1a.-e.	Requirements for in flight modification capabilities	Y
5.2.2.1a.-f.	Requirements for real-time	Y
5.2.2.1a.-g.	Requirements for security	N
5.2.2.1a.-h.	Quality requirements	Y
5.2.2.2a.	System and software observability requirements	Y
5.2.2.3a.	HMI requirements	N
5.2.3.1a.	Verification and validation process requirements	N
5.2.3.2a.	Validation requirements and scenario	Y
5.2.3.3a.	Installation an acceptance requirements at the operational and maintenance sites	N
5.2.4.1a.	Association of requirements to versions	N
5.2.4.1b.	Delivery content and media	N
5.2.4.2a.	System level integration support requirements	Y
5.2.4.3a.	External interface requirements specification	Y
5.2.4.4a.	System database content and allowed operational range	N
5.2.4.5a.	Design and development constraints	Y
5.2.4.6a.	OBCP requirements	N
5.2.4.7a.	Requirements for 'software to be reused'	Y
5.2.4.8a.	Software safety and dependability requirements	Y
5.2.4.9a.	Format and delivery medium of exchanged data	Y
5.2.5a.	SRR	Y

5.3.2.1a.	Software life cycle definition	N
5.3.2.1b.	Software life cycle definition	N
5.3.2.1c.	Development strategy, standards, techniques, development and testing environment	Y
5.3.2.1d.	Software life cycle definition	N
5.3.2.2a.	Identification of interface between development and maintenance	N
5.3.2.3a.	Software procurement process documentation and implementation	N
5.3.2.4a.	Automatic code generation management	N
5.3.2.4b.	Automatic code generation management	N
5.3.2.4c.	Automatic code generation management	N
5.3.2.4d.	Automatic code generation management	N
5.3.2.4e.	Automatic code generation configuration management	N
5.3.2.5a.	Changes to baseline procedures	N
5.3.3.1a.	Joint review reports	Y
5.3.3.2a.	Software project reviews included in the software life cycle definition	N
5.3.3.2b.	Review Plan	Y
5.3.3.3a.	Software technical reviews included in the software life cycle definition	N
5.3.3.3b.	Technical reviews process	N
5.3.3.3c.	Software technical reviews included in the software life cycle definition	N
5.3.4.1a.	Approved requirements baseline	Y
5.3.4.2a.	Approved technical specification and interface, architecture and plans	Y
5.3.4.2b.	Approved technical specification and interface	Y
5.3.4.3a.	Approved design definition file and design justification file	N
5.3.4.3b.	Approved detailed design, interface design and budget	Y
5.3.4.4a.	Qualified software product	N
5.3.4.5a.	Accepted software product	Y
5.3.5.1a.	Confirmation of readiness of test activities For validation and acceptance test activities only	Y
5.3.5.2a.	Approved test results For validation and acceptance test activities only	N

5.3.6.1a.	Flight software review phasing	N
5.3.6.1b.	Flight software review phasing	N
5.3.6.2a.	Ground software review phasing	N
5.3.7.1a.	Interface management procedures	N
5.3.8.1a.	Technical budgets and margin philosophy for the project	Y
5.3.8.2a.	Technical budgets and margin computation	N
5.3.9.1a.	E40 compliance matrix	N
5.3.9.2a.	E40 compliance matrix	N
5.4.2.1a.-a.	Functional and performance specifications, including hardware characteristics, and environmental conditions under which the software item executes, including budgets requirements	Y
5.4.2.1a.-b.	Operational, reliability, safety, maintainability, portability, configuration, delivery, adaptation and installation requirements, design constraints	Y
5.4.2.1a.-c.	Software product quality requirements (see ECSS-Q-ST-80 clause 7.2)	Y
5.4.2.1a.-d.	Security specifications, including those related to factors which can compromise sensitive information	N
5.4.2.1a.-e.	Human factors engineering (ergonomics) specifications, following the human factor engineering process described in ECSS-E-ST-10-11	N
5.4.2.1a.-f.	Data definition and database requirements	Y
5.4.2.1a.-g.	Validation requirements	Y
5.4.2.1a.-h.	Interfaces external to the software item	Y
5.4.2.1a.-i.	Reuse requirements	Y
5.4.2.2a.	Specifications for in flight software modifications	Y
5.4.2.3a.	Software logical model	Y
5.4.2.3b.	Software logical model method	Y
5.4.2.3c.	Behavioural view in software logical model	Y
5.4.2.4a.	SWRR	N
5.4.3.1a.	Software architectural design	Y
5.4.3.2a.	Software architectural design method	Y
5.4.3.3a.	Computational model	Y
5.4.3.4a.	Software behaviour	Y
5.4.3.5a.-a.	Preliminary external interfaces design	Y

5.4.3.5a.-b.	Preliminary internal interfaces design	Y
5.4.3.6a.	Software intended for reuse - justification of methods and tools	Y
5.4.3.6b.	Software intended for reuse - evaluation of reuse potential	Y
5.4.3.6c.	Software architectural design with configuration data	Y
5.4.3.7a.	Justification of reuse with respect to requirements baseline	Y
5.4.3.8a.	Software integration strategy	N
5.4.4a.	PDR	Y
5.5.2.1a.	Software components design documents	Ye
5.5.2.1b.	Software components design documents	Ye
5.5.2.1c.	Software components design documents	Y
5.5.2.2a.-a.	External interfaces design (update)	Y
5.5.2.2a.-b.	Internal interfaces design (update)	Y
5.5.2.3a.-a.	Software static design model	Y
5.5.2.3a.-b.	Software dynamic design model	Y
5.5.2.3a-.c.	Software behavioural design model	Y
5.5.2.4a.	Software design method	N
5.5.2.5a.	Real-time software dynamic design model	Y
5.5.2.5b.	Real-time software dynamic design model	Y
5.5.2.5c.	Real-time software dynamic design model	Y
5.5.2.5d.	Real-time software dynamic design model	Y
5.5.2.5e.	Real-time software dynamic design model	Y
5.5.2.6a.	Software behavioural design model techniques	N
5.5.2.7a.	Compatibility of real-time design methods with the computational model	Y
5.5.2.8a.	Software user manual	Y
5.5.2.9a.	Software unit test plan	N
5.5.2.10a.	DDR	N
5.5.3.1a.-a.	Software component design documents and code (update)	Y
5.5.3.1a.-b.	Software configuration file - build procedures	Y
5.5.3.2a.-a.	Software component design document and code (update)	Y
5.5.3.2a.-b.	Software unit test plan (update)	N
5.5.3.2b.-a.	Software component design document and code (update)	Y
5.5.3.2b.-b.	Software unit test reports	N

5.5.3.2c.	Software unit test reports	N
5.5.4.1a.	Software integration test plan (update)	N
5.5.4.2a.	Software integration test report	N
5.6.2.1a.	Software validation plan - validation process identification	Y
5.6.2.1b.	Software validation plan - methods and tools	Y
5.6.2.1c.	Software validation plan - effort and independence	N
5.6.2.2a.	Independent software validation plan - organization selection	N
5.6.2.2b.	Independent software validation plan - level of independence	N
5.6.3.1a.	Software validation specification with respect to the technical specification	Y
5.6.3.1b.	Software validation specification with respect to the technical specification	Y
5.6.3.1c.	Software validation specification with respect to the technical specification	Y
5.6.3.2a.	Software validation report with respect to the technical specification	Y
5.6.3.3a.	Software user manual (update)	Y
5.6.3.4a.	CDR	Y
5.6.4.1a.	Software validation specification with respect to the requirements baseline	Y
5.6.4.1b.	Software validation specification with respect to the requirements baseline	Y
5.6.4.1c.	Software validation specification with respect to the requirements baseline	Y
5.6.4.2a.	Software validation report with respect to the requirements baseline	Y
5.6.4.2b.	Software validation report with respect to the requirements baseline	Y
5.6.4.3a.	Software user manual (update)	Y
5.6.4.4a.	QR	N
5.7.2.1a.-a.	Software product	Y
5.7.2.1a.-b.	Software release document	N
5.7.2.2a.	Training material	N
5.7.2.3a.	Installation procedures	Y
5.7.2.4a.	Installation report	N

5.7.2.4b.	Installation report	N
5.7.2.4c.	Installation report	N
5.7.2.4d.	Installation report	N
5.7.3.1a.	Acceptance test plan	Y
5.7.3.2a.	Acceptance test report	Y
5.7.3.3a.	Software product	Y
5.7.3.4a.	Joint review report	Y
5.7.3.4b.	Joint review report	Y
5.7.3.5a.	Traceability of acceptance tests to the requirements baseline	Y
5.7.3.6a.	AR	Y
5.8.2.1a.	Software verification plan - verification process identification	Y
5.8.2.1b.	Software verification plan - software products identification	Y
5.8.2.1c.	Software verification plan - activities, methods and tools	Y
5.8.2.1d.	Software verification plan - organizational independence, risk and effort identification	N
5.8.2.2a.	Independent software verification plan - organization selection	N
5.8.2.2b.	Independent software verification plan - level of independence	N
5.8.3.1a.	Requirements baseline verification report	N
5.8.3.2a.-a.	Requirements traceability matrices	Y
5.8.3.2a.-b.	Requirements verification report	N
5.8.3.3a.-a.	Software architectural design to requirements traceability matrices	Y
5.8.3.3a.-b.	Software architectural design and interface verification report	N
5.8.3.4a.-a.	Detailed design traceability matrices	N
5.8.3.4a.-b.	Detailed design verification report	N
5.8.3.5a.-a.	Software code traceability matrices	N
5.8.3.5a.-b.	Software code verification report	N
5.8.3.5b.	Code coverage verification report	N
5.8.3.5c.	Code coverage verification report	N
5.8.3.5d.	Code coverage verification report	N

5.8.3.5e.	Code coverage verification report	N
5.8.3.5f.	Robustness verification report	Y
5.8.3.6a.-a.	Software unit tests traceability matrices	N
5.8.3.6a.-b.	Software unit testing verification report	N
5.8.3.7a.	Software integration verification report	N
5.8.3.8a.-a.	Traceability of the requirements baseline to the validation specification	Y
5.8.3.8a.-b.	Traceability of the technical specification to the validation specification	Y
5.8.3.8b.-a.	Validation report evaluation with respect to the technical specification	Y
5.8.3.8b.-b.	Validation report evaluation with respect to the requirements baseline	Y
5.8.3.9a.	Complement of validation at system level	N
5.8.3.10a.	Software documentation verification report	N
5.8.3.11a.	Schedulability analysis	Y
5.8.3.11b.	Schedulability analysis (update)	Y
5.8.3.11c.	Schedulability analysis (update)	Y
5.8.3.12a.	Technical budgets - memory and CPU estimation	Y
5.8.3.12b.	Technical budgets (update) - memory and CPU estimation	Y
5.8.3.12c.	Technical budgets (update) - memory and CPU calculation	Y
5.8.3.13a.	Software behaviour verification	Y
5.8.3.13a.	Software behaviour verification	Y
5.8.3.13b.	Software behaviour verification	Y
5.9.2.1a.	Software operation support plan - operational testing specifications	N
5.9.2.2a.	Software operation support plan - plans and procedures	N
5.9.2.3a.	Software operation support plan - procedures for problem handling	N
5.9.3.1a.	Operational testing results	Y
5.9.3.2a.	Operational testing results	Y
5.9.3.3a.	Software product	Y
5.9.4.1a.	Software operation support performance	N
5.9.4.2a.	Problem and nonconformance report	N
5.9.5.1a.	User's request record - user's request and subsequent actions	N

5.9.5.1b.	User's request record - user's request and subsequent actions	N
5.9.5.2a.	User's request record - actions	N
5.9.5.2b.	User's request record - actions	N
5.9.5.2c.	User's request record - actions	N
5.9.5.3a.	User's request record - work around solution	N
5.9.5.3b.	User's request record - work around solution	N
5.10.2.1a.	Maintenance plan - plans and procedures	N
5.10.2.1b.	Maintenance plan - applicability of development process procedures, methods, tools and standards	N
5.10.2.1c.	Maintenance plan - configuration management process	N
5.10.2.1d.	Maintenance plan - problem reporting and handling	N
5.10.2.1e.	Problem and nonconformance report	N
5.10.2.2a.	Maintenance plan - long term maintenance solutions	N
5.10.3.1a.	Modification analysis report and problem analysis report	N
5.10.3.1b.	Modification analysis report and problem analysis report	N
5.10.3.1c.	Modification analysis report and problem analysis report	N
5.10.3.1d.	Modification analysis report and problem analysis report	N
5.10.3.1e.	Modification approval	N
5.10.4.1a.	Modification documentation	N
5.10.4.2a.	Modification documentation	N
5.10.4.3a.	Modification documentation	N
5.10.4.3b.	Modification documentation	N
5.10.4.3c.	Modification documentation	N
5.10.4.3d.	Modification documentation	N
5.10.4.3e.	Modification documentation	N
5.10.5.1a.	Joint review reports	N
5.10.5.2a.	Baseline for changes	N
5.10.6.1a.	Migration plan	N
5.10.6.2a.	Migration plan	N
5.10.6.3a.	Migration plan	N
5.10.6.4a.	Migration plan	N
5.10.6.5a.	Migration notification	N
5.10.6.5b.	Migration notification	N

5.10.6.6a.	Post operation review report	N
5.10.6.6b.	Post operation review report	N
5.10.6.7a.	Migration plan	N
5.10.7.1a.	Retirement plan	N
5.10.7.2a.	Retirement notification	N
5.10.7.3a.	Retirement plan	N
5.10.7.4a.	Retirement plan	N