# MADES


**Model-based methods and tools for Avionics and surveillance embeddeD SystEmS**

FP7-ICT-2007 248864

# D1.1 Requirements Specification


| | |
|---|---|
| Work package: | WP1 |
| Lead participant: | EADS |
| Editor: | Gundula Blohm |
| Authors: | Gundula Blohm, Alessandra Bagnato, Stefano Genolini |
| Reviewers: | Richard Paige, Matteo Rossi |
| Date: | 30-07-2010 |
| Version: | 1.0 |



| Dissemination level | | |
|---|---|---|
| **PU** | Public | X |
| **PP** | Restricted to other program participants (including Commission Services) | |
| **RE** | Restricted by a group specified by the consortium (including Commission Services) | |
| **CO** | Confidential, only for members of the consortium (including Commission Services) | |

## The MADES Consortium

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| | | |
|---|---|---|
|  | **TXT e-solutions**<br>Via Frigia 27<br>20126 Milano<br>Italy | **Contact:** Ms. Alessandra Bagnato<br>E-mail: alessandra.bagnato@txt.it<br>Phone: +39 2 25771.725<br>Address: Via al Ponte Reale 5<br>16124 Genova, Italy |
|  | **SOFTEAM**<br>21 avenue Victor Hugo<br>75016 Paris<br>France | **Contact:** Dr. Andrey Sadovykh<br>E-mail: andrey.sadovykh@softeam.fr<br>Phone: +33.1.30.12.18.57 |
|  | **University of York**<br>Heslington<br>York YO10 5DD<br>United Kingdom | **Contact:** Prof. Richard Paige<br>E-mail: paige@cs.york.ac.uk<br>Phone: +44 1904437222 |
|  | **Politecnico di Milano**<br>via Golgi, 42<br>20133 Milano<br>Italy | **Contact:** Prof. Luciano Baresi<br>E-mail: baresi@elet.polimi.it<br>Phone: +39 02 2399 3638 |
|  | **The Open Group**<br>Avenue du Parc de Woluwe 56<br>B-1160 Brussels<br>Belgium | **Contact:** Scott Hansen<br>E-mail: s.hansen@opengroup.org<br>Phone: +32 2 6751136 |
|  | **EADS Deutschland**<br>Woerthstrasse 85<br>89077 Ulm<br>Germany | **Contact:** Ms. Gundula Blohm<br>E-mail: gundula.o.blohm@eads.com<br>Phone: +49 731 392 3757 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# Table of Contents



# List of Tables



The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# 1 Introduction

## 1.1 Role of this deliverable

This document provides a basis to understand the requirements for the tools to be developed within MADES. These tools are expected to provide support for avionics and surveillance embedded software development.

The UML profile MARTE shall be implemented and where needed extended to fulfill these requirements.

As embedded software is closely coupled to hardware, additional system modeling aspects have to be taken in account. In this context a system is defined as a technical system, consisting of both, hardware and software entities.

## 1.2 Structure of this document

Chapter 2 outlines the MADES identified tool usage use cases.

Chapter 3 outlines the identified requirements for the various MADES tools.

The requirements are derived by the current situation in EADS and TXT

In addition, some requirements are derived from the "Description of Work" and the "tool usage use cases" in Chapter 2.

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

## 1.3 Relationship to other MADES deliverables

? D1.4 MADES Integrated Tool Set - Initial Version

? D1.6 MADES Integrated Tool Set - Final Version

? and the parts of these which are developed in WP2, WP3 and WP4.

As case studies in WP5 are used to demonstrate fulfillment of the requirements,

? D5.1 Specification of the evaluation criteria of technical WPs

is directly linked to this document.

## 1.4 Contributors

The main contributors of this deliverable are EADS and TXT. All partners have provided ideas and feedback to the tool usage use cases and requirements; additionally, the document has undergone review by indirect contributors as part of the MADES quality assurance process.

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# 2 Tool Usage use cases

In order to achieve a common understanding of the MADES business and technological aspects, that is what problems the target system with its environment shall solve or what functionality it should implement, we have created a set of practical use case scenarios. The tool usage use cases explain how the target system will be used and are the main source for deriving requirements. The way they have been created is described by our Tool Usage use cases handling process found in Appendix A. Note that these scenarios have been created in a broad and non-limiting way in order to encourage innovation. The MADES partners have different backgrounds and expertise within this field, and by collecting ideas from everyone we have come up with a set of tool usage use cases. The derived requirements in section 3 state more clearly what we plan to implement and achieve within this project.

| **Scenario 1: Browse MADES Component Repository** | |
|---|---|
| **Description** | Developer connects to the MADES Component Repository Web Interface |
| **Workflow** | 1. Developers read the MADES approach guide to learn about how to benefit from MADES<br>2. Developers navigate through the MADES Component Repository content to learn about MADES MARTE Components (A Component is defined as self sufficient piece of information).<br>3. Developers download MADES MARTE Components to use them actively in their development process.<br>4. Developers connect to the MADES Component Repository to update it with their latest MARTE Component developed. |
| **Actors** | Developers. Comp Repository |
| **Outcome** | The MADES Component Repository helps organizing and using MARTE developed components. The MADES Component Repository becomes an important source for learning for Avionic and embedded systems software developers, and it helps software developers produce more efficiently. |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 2: Create a model for the Component Repository** | |
|---|---|
| **Description** | Upload new MARTE Components into MADES Component Repository |
| **Workflow** | 1. Developers model a formal description of the MARTE component.<br>2. A developer models/adds properties information.<br>3. Developers upload model into Component Repository.<br>4. The modeling tool (Modelio) is used to connect to the upload interface of the Component Repository<br>5. MARTE Model Component is uploaded to the Component Repository |
| **Alternate workflow** | 1b, 2b, 3b. Update existing model to MADES Component Repository<br>4b. MADES on line Component Repository is not available, model is stored locally for later upload |
| **Actors** | Developers, modeling tool |
| **Outcome** | Developers create and share MARTE Component models that can be used by developers to develop more efficiently Avionic and embedded systems software. |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 3: Search for MADES MARTE Component via Component Repository web interface** | |
|---|---|
| **Description** | A developer is currently creating a model and wants to check the Component repository for similar MARTE Components already developed (to reduce the chance of duplicate and inconsistent models, and to learn from others)<br><br>Through the component repository web interface developers can specify their search criteria, such as which MARTE components models to search for.<br><br>The web interface then connects to MADES Component repository where the search is performed. The result of the search is presented to the developer shortly after. The developer can then study the relevant MARTE Components in more detail, and select which component to link to or modify from within the modeling tool (e.g. Modelio). All relevant models can be downloaded directly to his/her modeling tool. |
| **Workflow** | 1. Developer specifies search criteria for the MARTE Component he/she is looking for<br>2. Component repository web interface connects to the search interface of the MADES Component repository with specified search criteria<br>3. Search is performed on the MADES Component repository<br>4. The result of the search is returned to the Developers<br>5. Developer selects the MARTE Component he/she wants to examine closer<br>6. The MARTE Components of interest are imported into modeling tool |
| **Alternate workflow** | 1b. Tool searches based on model<br><br>2b. MADES on line Component Repository not available<br><br>3b. No models fit the search criteria<br><br>5b. The developer does not want to download any model |
| **Actors** | Developers, modeling tool |
| **Outcome** | Developer gets an overview of relevant MARTE Components and can link to those he/she finds being most relevant for the model she is currently modeling within her modeling tool. |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 4: MADES -to- ZOT-to- MADES Transformation** | |
|---|---|
| **Description** | A developer wants to verify a MADES model using the ZOT verification tools |
| **Workflow** | 1. Developer builds a model in MADES<br>2. Developer chooses to verify a set of model properties using Zot<br>3. ETL is used to transform the model under consideration to the Zot representation<br>4. Zot finds no errors<br>5. The modeling tool provides the appropriate message to the developer |
| **Alternate workflow** | 1. Developer builds a model in MADES which includes requirements of the runtime behavior of the system (e.g. timing properties, power requirements)<br>2. Developer chooses to verify a set of model properties using Zot<br>3. ETL is used to transform the model under consideration to the Zot representation<br>4. Zot finds errors<br>6. Zot generates a textual counterexample<br>7. The tool parses the textual counter example and generates a counterexample model<br>8. The counterexample model is transformed back to MADES using ETL<br>9. The counterexample model represented in MARTE is presented to the developer<br>10. Using the counterexample and associated traceability information, the designer can determine which part of the system is failing to meet the specification and redesign either the hardware or software accordingly. |
| **Actors** | Developers, modeling tool, Zot, Epsilon platform |
| **Outcome** | Developer verifies particular properties of a model. If the verification fails, a counterexample is presented to the developer |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 5: Developers verify time-related properties described through behavioral diagrams** | |
|---|---|
| **Description** | A developer wants to check some temporal properties of a MADES model using a formal verification tool (e.g., Zot) |
| **Workflow** | 1. Developer describes system behavior using UML diagrams (e.g., sequence diagrams) annotated with MADES/MARTE stereotypes; in particular, annotations regarding timing properties, expressed in terms of clocks, are added to the diagram.<br>2. Using the MADES formal semantics of behavioral diagrams, the MADES diagrams defined by the developer are translated into a formal model suitable to be formally verified through an automatic tool.<br>3. Developer chooses properties to be verified through a dialog box (e.g. maximum distance between two events is T), and then instantiates the property on the behavioral diagram (e.g., decides which two events of the diagram should be separated by at most T).<br>4. Formal verification tool is invoked to check whether the desire property holds for the modeled system or not.<br>5. Verification tool finds no errors, and reports that the property is verified. |
| **Alternative workflow** | Steps 1, 2, 3, 4 as before<br>5. Verification tool determines that the desired property does not hold for the modeled system, and finds a counterexample<br>6. The counterexample is reported back to the developer in the form of a sequence diagram annotated with MARTE/MADES timing stereotypes. |
| **Actors** | Developers, modeling tool, formal verification tool. |
| **Outcome** | Developer verifies particular properties of a model. If the verification fails, a counterexample is presented to the developer |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 6: Developers simulate designed system in conjunction with a model of the environment with which it interacts** | |
|---|---|
| **Description** | A developer wants to validate the design of the system against a model of the environment with which it interacts. |
| **Workflow** | 1. Developer describes the behavior of the designed system through a collection of MADES behavioral diagrams.<br>2. Developer provides a model of the environment with which the designed system interacts through the Modelica language, and associates the Modelica model to elements in the MADES diagrams.<br>3. The formal semantics of MADES diagrams is used to create a formal model of the behavior of the designed system.<br>4. The formal model of MADES diagrams is used to produce example traces of the designed system; the Modelica model is used to simulate how the environment evolves in relation to the execution of the designed system.<br>5. The simulation tool that combines system and environment traces determines that the simulated behavior of the environment is compatible with the desired one, otherwise it stops when the compatibility is lost. |
| **Actors** | Developers, modeling tool, simulation tool. |
| **Outcome** | Developer validates the designed system if the simulation tool shows compatibility between the traces of the designed system and of its environment; otherwise it leads the simulation to a state where compatibility is lost. |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 7: MADES Hardware Design Space Exploration** | |
|---|---|
| **Description** | An engineer wants to experiment with different hardware architectures |
| **Workflow** | 1. The engineer builds behavioral and structural models of the software in MADES as well as models of the hardware architecture<br>2. The engineer generates platform agnostic code using the code generation facilities of MADES tool.<br>3. Compile time virtualization (CTV) facility of the tool is used to produce platform specific code for the targeted platform<br>4. The performance of the system is tested and it is not found to be satisfactory.<br>5. The engineer changes the hardware architecture model and the mappings of the platform agnostic code to the new architecture without changing the platform agnostic code.<br>6. CTV is used again to generate platform-specific code for the new platform description<br>7. The performance of the system is tested again |
| **Alternate workflow** | 1. Engineer builds behavioral and structural models of the system in MADES as well as models of the hardware architecture<br>2. The engineer generates platform agnostic code using the code generation facilities of MADES tool.<br>3. Compile time virtualization (CTV) facility of the tool is used to produce platform specific code for the targeted platform<br>4. The performance of the system is tested and it is found to be satisfactory. |
| **Actors** | Developers, modeling tool, CTV tool, Epsilon platform |
| **Outcome** | Engineers experiment with different hardware architectures in order to find the optimal one. |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 8: Requirementscoverage analysis** | |
|---|---|
| **Description** | An engineer wants to verify that every requirement corresponds to at least one design component |
| **Workflow** | 1. The engineer builds the design of the system using the MADES modeling language..<br>2. After the design is complete the engineer wants to verify that all the requirements correspond to at least one design component<br>3. The corresponding coverage analysis facility of the tool is used.<br>4. “Orphan” requirements are detected<br>5. The engineer refines the initial design by taking into consideration the “orphan” requirements. |
| **Alternate workflow** | 1. The engineer builds the design of the system using the MADES modeling language..<br>2. After the design is complete the engineer wants to verify that all the requirements correspond to at least one design component<br>3. The corresponding coverage analysis facility of the tool is used.<br>4. All the requirements correspond to at least one design component<br>5. Engineers continue with the implementation of the design. |
| **Actors** | Developers, modeling tool, Epsilon platform |
| **Outcome** | All requirements are taken into consideration during the design phase. |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| **Scenario 9: First encounter with MADES Solution** | |
|---|---|
| **Description** | Developer connects to the MADES Integrated Environment for the first time |
| **Workflow** | Developers reads the MADES approach guide to learn about how to benefit from MADES<br>Developers start to use the MADES Integrated Environment |
| **Actors** | Developers. MADES Integrated Environment |
| **Outcome** | The MADES Approach guide becomes an important source for learning for Avionic and embedded systems software developers, and it helps software developers to learn to use the MADES Integrated Environment |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# 3 Requirements

MADES Requirements for technical workpackages and WP1 integration workpackage are described with *ID, title, description, type, source scenario, author, priority.* A subset of these is included in this document:

- **ID:** Every requirement has a unique ID denoted as *R <.sequence number> (e.g. R.1 is the first requirement).*
- **Title:** The title is unique and briefly indicates what the requirement is about.
- **Description:** A short explanation of the requirement. The level of detail is not at a technical level, leaving room for design and implementation choices.
- **Source:** Every requirement has a source, (one or more scenarios) that put the requirement into the project context and supplements the description.
- **Priority:** A requirement has a priority ranging from 1 to 2, where 1 is considered mandatory and 2 is considered as nice to have.

## 3.1 Requirements related to Integrated Environment

| Req ID | Title | Description | Source / Rationale | Author | Priority |
|---|---|---|---|---|---|
| R 1.1 | Introduction in Toolset | There shall be a Guide explaining how to use the Integrated Environment with a small example project. | Usability | EADS | 1 |
| R 1.2 | Integration of Toolset | Modelling and Verification shall be integrated in a single framework (the user should not perceive change of environment). | Usability | EADS<br>TXT | 1 |
| R 1.3 | Applicability of Toolset | TXT and EADS workflows shall be supported. | TXT and EADS as is situations | EADS<br>TXT | 1 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

## 3.2 Requirements related to Component Repository

| Req ID | Title | Description | Source | Author | Priority |
|---|---|---|---|---|---|
| R 2.1 | Introduction in Component Repository | There shall be a Guide explaining how to do search for MARTE Components inside the MADES Component Repository. | Scenario 1 | TXT | 1 |
| R 2.2 | Component Repository API specification | An API for external modeling tool connection shall be defined. | Scenario 1,2,3 | TXT | 1 |
| R 2.3 | Provision of Components | The Component Repository shall provide previously stored components to modeling tools. | Scenario 3 | EADS | 1 |
| R 2.4 | Provision of Components Library | The Component Repository shall provide previously stored sets of components to modeling tools. | Scenario 3 | EADS | 2 |
| R 2.5 | Storage of Components | The Component Repository shall provide means to modeling tools to store components. | Scenario 2 | EADS | 1 |
| R 2.6 | Storage of Components Library | The Component Repository shall provide means to modeling tools to store sets of components. | Scenario 2 | EADS | 2 |
| R 2.7 | Search for Components | The Component Repository shall provide means to modeling tools to search for components. | Scenario 3 | EADS | 1 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

## 3.3 Requirements related to Modeling Tool

| Req ID | Title | Description | Source | Author | Priority |
|---|---|---|---|---|---|
| R 3.1 | Domain Support | The Modeling Tool shall support developers in creation and analysis of models for avionics and surveillance systems. | DoW | TXT | 1 |
| R 3.2 | Support for software design standard constraints | The Modeling Tool shall provide means to formulate software design constraints and verify their fulfilment. | TXT as is situation | TXT | 2 |
| R 3.3 | Support for different stakeholders | Depending on the role of the user, different parts of the model shall be editable / viewable. | EADS as is situation | EADS | 2 |
| R 3.4 | Top-Down Engineering | The Modeling Tool shall provide a model structure which supports different abstraction levels. | EADS as is situation | EADS | 1 |
| R 3.5 | Navigation | It shall be easy to navigate through different abstraction levels (zoom in and zoom out). | EADS as is situation | EADS | 1 |
| R 3.6 | Refinement of Elements | The modeling tool shall provide means to create multiple refinements of model elements.<br>E.g. a Use Case will be refined for external and internal view. | EADS as is situation | EADS | 1 |
| R 3.7 | Navigation through refinements | It shall be easy to find and navigate to different refinements of a model element. | EADS as is situation | EADS | 1 |
| R 3.8 | Classification of refinements | It shall be possible to classify elements or refinements as approved or rejected. | EADS as is situation | EADS | 1 |
| R 3.9 | Status of requirements | It shall be possible to model the lifecycle state of a requirement. | EADS as is situation | EADS<br>TXT | 1 |
| R 3.10 | Creation and Maintenance of components | The Modeling Tool shall support the creation and maintenance of components to be used in system models. | Scenario 2 | TXT | 1 |
| R 3.11 | Usage of components | The Modeling Tool shall support the developer to compose an avionic and surveillance system model using previously stored components. | Scenario 3 | TXT | 1 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| Req ID | Title | Description | Source | Author | Priority |
|---|---|---|---|---|---|
| R 3.12 | User friendly notation | The Modeling notation shall be understandable also by users which are not experts in Avionic and Surveillance domain. | TXT as is situation | TXT | 1 |
| R 3.13 | UML subset | The already used subset of UML2 shall be supported. | EADS as is situation | EADS | 1 |
| R 3.14 | UML extensions | There shall be suitable elements for modeling system interfaces with their properties (logical & physical). | EADS as is situation | EADS | 1 |
| R 3.15 | Interface Delegation | Delegation from Component Interface to Composite Interface or Activity Parameter shall be supported. | EADS as is situation | EADS | 1 |
| R 3.16 | Data Modeling | Means shall be provided to model exact (bitwise) data description. | EADS as is situation | EADS | 1 |
| R 3.17 | Resource Modeling | The Modeling Tool shall allow to specify information about timing budget and resource usage. | Discussion | EADS TXT POLIMI | 1 |
| R 3.18 | Dependency Awareness | The Modeling Tool shall be able to detect dependencies between behaviors (Components involved in several scenarios). | Discussion | EADS TXT | 2 |
| R 3.19 | Time Modeling | The Modeling Tool shall be able to model time using MARTE | Discussion | TXT | 2 |
| R 3.20 | Processes Types | The Modeling Tool shall be able to model processes type and duration (periodic or statistic) and communications mechanisms between different entities (e.g. synchronous or a asynchronous). | Discussion | TXT | 1 |
| R 3.21 | Processing Capacity | Modeling capacity of components (e.g. how many requests can be received). | Discussion | TXT | 1 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

## 3.4 Requirements related to Verification Tool

As Verification Tools are not part of the actual workflow in place at TXT and EADS, there are not that concrete requirements as for the Modeling Tool.

In this work package the contributors are expected to present features, which can then be evaluated in the case studies.

| Req ID | Title | Description | Source | Author | Priority |
|---|---|---|---|---|---|
| D 4.1 | Usability | The developers can easily interact with the tool and can select the properties of interest. | Discussion | TXT | 1 |
| D 4.2 | Properties | Developers can define the properties to be verified and receive a response. | Discussion | TXT | 1 |
| D 4.3 | Dependencies | Verification tool should be able to interpret dependencies specified at modeling level and verify timing properties (e.g. scheduling properties). | Discussion | TXT | 1 |

## 3.5 Requirements related to Simulation Tool

As Simulation Tools are not part of the actual workflow in place at TXT and EADS, there are not that concrete requirements as for the Modeling Tool.

In this workpage the contributors are expected to present features, which can then be evaluated in the case studies.

| Req ID | Title | Description | Source | Author | Priority |
|---|---|---|---|---|---|
| D 5.1 | Usability | The developers have suitable interfaces to launch the simulation of their model. | Discussion | TXT | 1 |
| D 5.2 | Behavior | The simulation of the behavior of the system should be possible. | Discussion | TXT | 1 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

## 3.6 Requirements related to Transformations Tool

| Req ID | Title | Description | Source | Author | Priority |
|---|---|---|---|---|---|
| R 6.1 | Tracking of requirements' changes | Track on each requirement change in the baselines. | TXT as is situation | TXT | 1 |
| R 6.2 | Documentation of requirements' changes | Requirement change should be documented saying which object and requirements it has affected (ECR= Engineering change request). | TXT as is situation | TXT | 1 |
| R 6.3 | Implementation Tracing | Each requirement must allow to trace operations that implement that requirement and its use cases (and vice versa). | TXT as is situation | TXT | 1 |
| R 6.4 | Trace Verification | A verification procedure to control that each requirement is linked to at least one operation (low level design) and one use case (high level design). | TXT as is situation | TXT | 1 |
| R 6.5 | Test Tracing | Each test sequence must be traced versus one or more requirements. | TXT as is situation | TXT | 1 |
| R 6.6 | Code Generation | The code generated should be 'readable' (otherwise code is not certifiable). | TXT as is situation | TXT | 1 |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# 4 References

/1/ OMG, UML Version v2.1.2, formal/07-02-05,
http://www.omg.org/spec/UML/2.1.2

/2/ OMG, UML Profile for MARTE
http://www.omg.org/cgi-bin/doc?ptc/ 09-11-02

/3/ IBM Rational DOORS
http://www-142.ibm.com/software/products/gb/en/ratidoor?S_TACT=none&S_CMP=none

/4/ Simulink - Simulation and Model-Based Design
http://www.mathworks.com/products/simulink

/5/ Artisan Real Time Studio ™, by Artisan Software Tools
http://www.artisansoftwaretools.com

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# Appendix A: Tool Usage Use Cases handling process

? All partners in the MADES project are expected to contribute with writing tool usage use cases (scenarios).

? Rough tool usage use cases or descriptions for the most important functionality or processes shall be presented according to the example given in Table 1.

? Tool usage use cases accepted are sent to WP1 for refinement.

? Textual descriptions are refined elaborated and consistency checked.

? UML Use Cases are modeled for each scenario.

? During the refinement process it is evaluated whether all aspects of the MADES project are covered by the scenarios or not. If not, WP1 may ask relevant partners for complementing scenario descriptions within certain areas.

| **Scenario X:** | |
|---|---|
| **Description** | |
| **Workflow** | |
| **Alternative workflow** | |
| **Actors** | |
| **Outcome** | |

**Table 1 Scenario template**

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

# Appendix B: Abbreviations

| | |
|---|---|
| ADC | Air Data Computer |
| AC&OUT | Acquisition & Output Module (I/O Interface Module) |
| ACSR | Active Control Structural Response |
| ADF | Automatic Direction Finder |
| AFCS | Automatic Flight Control System |
| AHRS | Attitude & Heading Reference System |
| AIM | Analog Input Module (I/O Interface Module) |
| AMMC | Aircraft & Mission Management Computer |
| AMS | Aircraft Management System |
| AMMCP | Aircraft & Mission Management Control Panel |
| API | Application Programming Interface |
| APU | Auxiliary Power Unit |
| APSW | Application Software |
| ARINC | Aeronautical Radio Inc. |
| A-RTS | Ada Runtime System |
| BC | Bus Controller |
| BM | Bus Monitor |
| CASPER | Communication and Survivability Processor Embedded Resource |
| CCC | Cross Channel Check |
| CDS | Cockpit Display System |
| CEC | Central Erase Command |
| CSCI | Computer Software Configuration Item |
| COMM BC | 1553B BUS Interface Module |
| COTS | Commercial Off-The-Shelf |
| DBU | Database Update |
| DF | Direction Finder |
| DID | Data Item Description |
| DIM | Digital Input Module (I/O Interface Module) |
| DME | Distance Measuring Equipment |
| DTC | Data Transfer Cassette |
| DTU | Data Transfer Unit |
| DVS | Doppler Velocity Sensor |
| EAS | Emergency Avionic System |
| EECU | Electronic Engine Control Unit |
| EGI | Embedded GPS & INS System |
| EGPWS | Enhanced Ground Proximity Warning System |
| EQSW | Equipment Software |
| FCC | Flight Control Computer |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.

| | |
|---|---|
| FGC | Fuel Gauging Computer |
| GPS | Global Positioning System |
| HMI | Human Machine Interface Software Component |
| HW | Hardware |
| HWCI | Hardware Configuration Item |
| I/O | Input/ Output |
| ICD | Interface Control Document |
| IFF | Identification Friend or Foe |
| ISDS | Ice & Snow Detection System |
| KAS | Kernel Address Space |
| LAN | Local Area Network |
| MCDU | Multifunction Display Unit |
| MCSP | Merlin Capability Sustainment Plus |
| N/A | Not Available |
| NVM | Non Volatile Memory |
| OPSW | Operational Software |
| OSD | Object Sequence Diagram |
| PSU | Power Supply Unit |
| RA | Radio Altimeter |
| RAIM | Receiver Autonomous Integrity Monitoring |
| RIPU | Rotor Ice Protector Unit |
| RIU | Remote Interface Unit |
| RT | Remote Terminal |
| RTOS | Real-time Operating System |
| RTS | Real-time System |
| SD | Sequence Diagram |
| SDD | Software Design Description |
| SRS | Software Requirements Specification |
| SS | System Specification |
| STLD | Software Top Level Design |
| SW | Software |
| TACAN | Tactical Airborne Navigation |
| TBD | To Be Defined |
| TSU | Touch Screen Unit |
| TVM | Transmission Vibration Monitoring |
| UCD | Use Case Diagram |
| UML | Unified Modeling Language |
| VAS | Virtual Address Space |
| VDAM | Vibration Data Acquisition Module |

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° 248864.