

Rafael Valiev
Gregory Yermolaev
Kasatkin Timur

Innopolis University

myCrash Variant

M e s s i r **User Manual**
- v 1.0 -

Based on IEEE Std 1063-2001 [\[1\]](#)

Friday 3rd February, 2017 - 23:01

Springer

Contents

1	Product information	5
1.1	Identification	5
1.2	Copyright	5
1.3	Trademark notices	5
1.4	Restrictions	5
1.5	Warranties	5
1.6	Contractual obligations	5
1.7	Disclaimers	5
1.8	Contact	5
2	Introduction	7
2.1	Scope	7
2.2	Purpose	7
2.3	Intended audience	7
2.4	<i>myCrashVariant</i>	7
2.4.1	Actors & Functionalities	8
2.4.2	Operating environment	9
2.5	Document structure	9
3	Usage Guide	11
3.1	Multi-procedures	12
3.1.1	MyMultiProcedure1	12
3.1.2	MyMultiProcedure2	12
3.1.3	MyMultiProcedure3	12
3.2	Mono-procedures	12
3.2.1	MyActor1	12
3.2.2	My-Actor2	12
4	Software operations	13
4.1	MyOperation	13
4.1.1	MyExample1	13
5	Error messages and problem resolutions	15
5.1	Error message 1	15
5.1.1	Problem identification	15
5.1.2	Probable cause	15
5.1.3	Corrective actions	15
A	Title of the appendix 1	17
A.1	My Section	17
A.1.1	My subSection	17
Glossary		19
References		21

List of Figures

Chapter 1

Product information

1.1 Identification

Include precise information of the software product like identification name (that you can include in the [Glossary](#)), list of parts that compose it (indicating identification numbers for each part). Specify the applicable operating environment(s), including version(s) of hardware, communications, and operating system(s).

1.2 Copyright

1.3 Trademark notices

1.4 Restrictions

Restrictions on copying or distributing the software and its associated documentation.

1.5 Warranties

1.6 Contractual obligations

1.7 Disclaimers

1.8 Contact

Information for contacting the issuing organization.

Chapter 2

Introduction

2.1 Scope

This section has to provide the scope of the user's manual document. In the following some opening statements to use when providing the information corresponding to this section.

This document provides minimum acceptable information for knowing how to use the software system *myCrash Variant*.

This document does not provide all details regarding *myCrash Variant*.

This document is not intended to provide information about how to connect, deploy, configure, or use any external device or third-party software system that is required for the correct functioning of *myCrash Variant*.

This document may be used with other documents provided by third-party companies to have an overall view and correct understanding of the environment and procedures where the software system *myCrash Variant* is aimed to be deployed and run.

2.2 Purpose

In this section you explain the purpose (i.e. aim, objectives) of the user's manual. In the following some examples of opening statements to be used in this section.

The purpose of this document is to give some minimum amount of information required for using the system.

This document defines such models as [Environment Model](#), [Concept Model](#), etc. It also defines several views such as [Deployment View](#), [Implementation View](#), etc.

2.3 Intended audience

Description of the categories of persons targeted by this document together with the description of how they are expected to exploit the content of the document.

2.4 *myCrash Variant*

Brief overview of the software application domain and main purpose.

2.4.1 Actors & Functionalities

2.4.1.1 Communication Company

A company that has the capacity to ensure communication of information between its customers and the *myCrash Variant* system.

Functionalities:

- deliver SMS about possible crisis got from witness or victim to the *myCrash Variant* 's phone number in form of alert
- be notified when handling alert by Coordinator of *myCrash Variant* system, sent by the Communication Company previously, is in progress
- transmit SMS messages from company that owns *myCrash Variant* system to any human having an SMS compatible device accessible using a phone number

2.4.1.2 Human

Any person who considers himself related to a car crash either as a witness, a victim or an anonymous person.

Functionalities:

- inform the *myCrash Variant* system about the crisis situation he detected by sending SMS with information about the crisis to some Communication Company which will send an alert to *myCrash Variant* system
- be notified that the ABC company has been informed about the situation
- be informed about the situation of the crisis he/she has related to as a victim or witness
- get QA survey (set of satisfaction single choice text questions with possible mark answer for each from 0 to 5) by SMS message when a crisis, associated with this Human, processed by *myCrash Variant* 's Coordinator. QA survey will be open (*myCrash Variant* system will wait answer from Human) for 5 hours.
- answer obtained QA survey by sending reply in form of corresponding mark-answers for each question separated by spaces

2.4.1.3 Coordinator

An employee of the company, owning *myCrash Variant* system, being responsible of handling one or several crisis.

Functionalities:

- authenticate in *myCrash Variant* system by providing valid (present in the system) login/password and, if login/password pair is valid, pressing on fingerprint scanner with his finger in special mobile app in the following 60 seconds
- observe alerts by their status (pending/valid/invalid)
- observe crises by their status (pending/handled/solved/closed)
- validate/invalidate pending alert (mark alert as real or not)
- handle crisis
- leave free text comment on crisis
- change crisis's status
- close crisis (make crisis's status equal to 'closed')

2.4.1.4 Administrator

An employee of the company, owning *myCrash Variant* system, being responsible of administrating the system.

Functionalities:

- authenticate in *myCrash Variant* system by providing valid (present in the system) login/password and, if login/password pair is valid, pressing on fingerprint scanner with his finger in special mobile app in the following 60 seconds

- add or delete coordinators from the system and its environment
- be notified about intrusions (when some person tried to authenticate as an Coordinator, but provided invalid (not stored in *myCrashVariant* system) login/password pair or his fingerprint scan didn't match one associated with corresponding (defined by login/password pair) coordinator)
- visualize satisfaction diagrams in form of eye chart showing relative amounts of each type of mark of for given QA survey question. The administrator can see such chart of question for all crises or for only specified one.

2.4.1.5 Creator

A technician who is installing the *myCrashVariant* system on the targeted deployment infrastructure.

Functionalities:

- install the *myCrashVariant* system
- define the values for the initial system's state
- define the values for the initial system's environment

2.4.1.6 Activator

A logical representation of the active part the *myCrashVariant* system. It represents an implicit stakeholder belonging to the system's environment that interacts with the iCrash system autonomously without the need of a external entity. It is usually used for representing time triggered functionalities.

Functionalities:

- communicate the current time to the system
- notify the administrator that some crisis are still pending for a too long time

2.4.2 Operating environment

Brief overview of the infrastructure on which the software is deployed and used.

2.5 Document structure

Information on how this document is organised and it is expected to be used. Recommendations on which members of the audience should consult which sections of the document, and explanations about the used notation (i.e. description of formats and conventions) must also be provided.

Chapter 3

Usage Guide

This section is aimed at describing the general use of the software, since it is **deployed, configured and run**.

This software is used by actors. These actors rely on the software to perform a set of business activities (called here procedures) aimed at reaching a particular goal.

These procedures are split in two groups:

- **Multi-procedures:** which are procedures at **summary** or **user-goal** level involving several active or pro-active actors. Each of these procedures aims at illustrating intertwined business activities required to be performed by the involved actors to reach the expected goal. Each business activity between the system and an actor must correspond to a **system operation** instance given with actual parameter values.
- **Mono-procedures:** which are procedures at **summary** or **user-goal** level involving only one active or pro-active actor. Each of these procedures aims at illustrating the required business activities an actor has to perform to reach the expected goal. Each business activity between the system and the actor must correspond to a **system operation** instance given with actual parameter values.

Each process has to be documented using the following textual description template [2] **BUT its content must be as low level as possible with actual values:**

Procedure: ProcessMissionOne
Scope: Crisis Management System (*CMS*)
Primary Actor: Coordinator John
Secondary Actor(s): FirstAidWorker Bob,
ExternalResourceSystem ERS
Goal: The intention of the Coordinator is to process mission with ID equal to 1.
Level: User-goal level
Main Success Scenario :
1. *John* instructs the *CMS* to process the mission with ID equal to 12.031005
2. *CMS* selects the internal worker *Bob* to execute the mission 12.031005
3. *CMS* instructs *Bob* to behave as *First Aid Worker (FAW)*
4. *Bob* informs the *CMS* of his arrival
5. *Bob* informs the *CMS* that he starts to execute the mission 12.031005
6. *Bob* informs the *CMS* that the mission 12.031005 outcome is “Mission completed”
Extensions :
2.a None internal worker can execute the mission
2.a.1 *CMS* sends a request for an external resource to the *ERS* actor instance
2.a.2 *ERS* informs *CMS* that the request can be processed
2.a.3 *ERS* informs *CMS* that *Bob* can now be selected as first aid worker
procedure continues at step 3

Remark-Processes presentation: processes should be introduced to the reader in a pedagogical manner. Thus, simple and common processes should be presented before than more complex and less utilised ones.

Remark-Graphical User Interfaces (GUIs): include GUIs screenshots to show the different stages of the process while its is performed by the actor(s).

3.1 Multi-procedures

3.1.1 *MyMultiProcedure1*

...

3.1.2 *MyMultiProcedure2*

...

3.1.3 *MyMultiProcedure3*

...

3.2 Mono-procedures

Mono-procedures must be grouped by actors.

3.2.1 *MyActor1*

3.2.1.1 MyProcedure1MyActor1

...

3.2.1.2 MyProcedure2MyActor1

...

3.2.2 *My-Actor2*

3.2.2.1 MyProcedure1MyActor2

...

3.2.2.2 MyProcedure2MyActor2

...

Chapter 4

Software operations

Explain each allowed software operations (i.e. an atomic unit of treatment, a service, a functionality) including a brief description of the operation, required parameters, optional parameters, default options, required steps to trigger the operation, assumptions upon request of the operation and expected results of executing such operation. Describe how to recognise that the operation has successfully been executed or abnormally terminated. The template given below (i.e. section 4.1 has to be used).

Group the operations devoted to the needs of specific actors. Common operations to several actors may be grouped and presented once to avoid redundancy.

4.1 MyOperation

The system operator creates and adds a new crisis to the system after being informed by a third party (citizen, organization) and selects a crisis handler for the crisis.

Parameters: Reporter Personal Information, Crisis Information, Crisis Handler

Precondition: The system operator is logged in and has received information from a reporter.

Post-condition: A new crisis has been added to the system and the new crisis has been assigned to a crisis handler, the Handler has received an automatic notification from the system.

Output messages: The selected Crisis Handler will be notified automatically once the crisis has been created.

Triggering:

1. From within the crisis management window fill out the required entries related to the personal information of the reporter such as name and phone number.
2. Fill out the entries related to the crisis type, impacted area, priority, description, GPS coordinates, address and finally choose a Crisis Handler from the combo box.
3. Click on the “Submit” button in and add the entry to the database.

4.1.1 MyExample1

Examples should illustrate the use of **complex operations**.

Each example must show how the actor uses the software operation under description to achieve (at least one of) its expected outcome.

It might be required to include GUI screenshots to illustrate the example.

Chapter 5

Error messages and problem resolutions

All known problems in using the software should be listed and explained in details using the structure presented below.

Contact information for reporting any problems (either with the software or this document) should be clearly indicated

5.1 Error message 1

5.1.1 Problem identification

A description explaining the meaning of the faced problem.

5.1.2 Probable cause

A description explaining the reasons why such a problem has been raised.

5.1.3 Corrective actions

Describe the required steps the actor should take to recover from such situation.

Appendix A

Title of the appendix 1

Here you write the context of the appendix, structuring such content in sections, sub-sections and sub-sub-sections, if needed.

An example of appendix is the flat presentation of all the graphical user interface screens. Each screen can be presented (identification symbol and description) and screens transition graph can be given.

A.1 My Section

Description of the section.

A.1.1 My subSection

A.1.1.1 My subSubSection

Glossary

Actor	An actor is a person, organization, or external system that plays a role in one or more interactions with the system.	8
Concept Model	the Model that describes the different types required to specify the software system. . . .	7
Deployment View	The physical view depicts the system from a system engineer's point-of-view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. For example, how many nodes are used and what is deployed on what node. A Deployment View is modelled as a UML Deployment Diagram.	7
Environment Model	the Model that describes the different actors supposed to interact with the software system.	7
Glossary	the description of terms that are likely unfamiliar to the audience. The glossary shall include an alphabetical list of terms and definitions. Documentation using abbreviations and acronyms unfamiliar to the audience shall include a list with definitions, which may be integrated with the glossary. Terms included in the glossary should also be defined on their first appearance in printed documentation. Here there is an example of how to include an expression into the glossary: Societics	5
Implementation View	This view describes the software system components. It focuses on software modules and subsystems. It describes the hierarchies or layers for components. This view is modelled as a UML Component Diagram.	7
Societics	Represents the fields of hardware/software systems used for the society extension.	17

References

1. IEEE: IEEE Standard for Software User Documentation. IEEE Std 1063-2001 (Dec 2001) 1–24
2. Armour, F., Miller, G.: Advanced Use Case Modeling: Software Systems. Addison-Wesley (2001)