



POLITECNICO
MILANO 1863

Computer Science and Engineering

Software Engineering 2

Academic year 2021-2022

Requirements Analysis and Specification
Document

DREAM

Data-dRiven PrEdictive FArMing in Telengana

Authors:

Elisa SERVIDIO	996387
Federica SURIANO	953085
Arslan ALI	971503

- -, 2021

Version -.0

Contents

List of Tables	3
List of Figures	4
1 Introduction	5
1.1 Purpose	5
1.1.1 Goals of the Application	7
1.2 Scope	8
1.2.1 World and Shared phenomena	8
1.3 Definitions, Acronyms, Abbreviations	8
1.3.1 Definitions	9
1.3.2 Acronyms	9
1.3.3 Abbreviations	9
1.4 Revision History	9
1.5 Reference Documents	9
1.6 Document Structure	9
2 Overall Description	10
2.1 Product Perspective	10
2.1.1 Scenarios	10
2.1.2 Class Diagrams	10
2.1.3 State Charts	10
2.2 Product functions	10
2.3 Users Characteristics	10
2.4 Assumptions, dependencies and constraints	10
2.4.1 Domain Assumptions	11
2.4.2 Dependencies	11
3 Specific Requirements	12
3.1 External Interface Requirements	12
3.1.1 User Interfaces	12
3.1.2 Hardware Interfaces	12
3.1.3 Software Interfaces	12
3.1.4 Communication Interfaces	12

3.2	Functional Requirements	12
3.2.1	Use Case Diagrams	12
3.2.2	Use Case Analysis	13
3.2.3	Sequence Diagrams	13
3.2.4	Requirements	13
3.2.5	Traceability Matrix	13
3.3	Performance Requirements	13
3.4	Design Constraints	13
3.4.1	Standards compliance	13
3.4.2	Hardware limitations	13
3.4.3	Any other constraint	13
3.5	Software System Attributes	13
3.5.1	Reliability	13
3.5.2	Availability	14
3.5.3	Security	14
3.5.4	Maintainability	14
3.5.5	Portability	14
4	Formal Analysis using Alloy	15
4.1	Alloy Model	15
4.1.1	Signatures	15
4.1.2	Facts	15
4.1.3	Assertions	15
4.1.4	Analysis Results	15
5	Effort Spent	16
	Bibliography	17

List of Tables

List of Figures

Chapter 1

Introduction

The Requirement Analysis and Specification Document (RASD) has the purpose of describing, to a wide range of potential readers, the system to be developed for the problem under consideration.

It contains the description of the scenarios, the corresponding use cases and the models describing requirements and specifications.

It focuses also on interactions between the system and the users, including the implied constraints and functionalities that are going to be implemented.

1.1 Purpose

Telangana is the 11th largest and the twelfth-most populated state in India with a geographical area of 112,077 km² and 35,193,978 residents (data from 2011).

The economy of Telangana is mainly driven by agriculture, a sector which plays a pivotal role in all India's economy: over 58% of rural households depend on it as the principal means of livelihood, 80% of whom are smallholder farmers with less than 2 hectares of farmland. More than a fifth of the smallholder farm households are below poverty.

Worldwide there are many threats to the agriculture sector. World population is estimated to reach 9.7 billion by 2050 (as per a recent UN estimate), therefore food demand is expected to increase anywhere between 59% to 98% by 2050 (source Harvard Business Review). Climate change is predicted to result in a 4%-26% loss in net farm income towards the end of the century. The COVID-19 pandemic has greatly exposed the vulnerabilities of marginalized communities, small holder farmers and the importance of building resilient food systems.

This calls for a revamp of the entire food supply chain to help bolster countries against shocks and challenges developing and adopting innovative methodologies and technologies.

For this reason, Telangana's government aims to design, develop and demonstrate anticipatory governance models for food systems using digital public goods and community-centric approaches to strengthen data-driven policy making in the state.

This will require the involvement of multiple stakeholders, from normal citizens to policy makers, farmers, market analysts, agronomists, etc.

To achieve this goal, Telangana wants to partner with IT providers with the aim of acquiring and combining data concerning: meteorological forecasts, agriculture production (types of products, produced amount per product, amount of water used by each farmer), humidity of soil and information obtained by the governmental agronomists who periodically visit the farms in the geographic area which they are assigned to.

Acquiring and combining such data, the software system DREAMS supports the work of three types of actors: policy makers, farmers, and agronomists.

DREAMS allows Telangana's policy makers to identify farmers who are performing well and those who are performing particularly badly. The first ones, especially the more resilient to meteorological adverse events, will receive special incentives and will be asked to provide useful best practices to the others. Moreover, the system will help policy makers to understand whether the steering initiatives carried out by agronomists with the help of good farmers produce significant results.

On the other hand, farmers are allowed to visualize data relevant to them based on their location and type of production, such as weather forecasts and personalized suggestions (i.e. specific crops to plant or specific fertilizers to use). They can insert in the system data about their production and any problem they face for which they can request for help and suggestions by agronomists and other farmers with whom they can also create discussion forums.

Eventually, agronomists, who are in charge of a certain area, can receive information about requests for help and answer them and they can visualize data concerning weather forecasts and the best performing farmers in the area. Furthermore, agronomists can visualize, update and confirm a daily plan to visit farms in the area, assuming that all farms must be visited at least twice a year, but those that are under-performing should be visited more often, depending on the type of problem they are facing.

1.1.1 Goals of the Application

- *G.1*: Allows policy makers to identify those farmers who are performing well
- *G.2*: Allows policy makers to identify those farmers who are performing badly
- *G.3*: Allows policy makers to verify the improvement of farmers who have been already helped by agronomist or good farmers
- *G.4*: Allows farmers to visualize data relevant to them based on their location and type of production
- *G.5*: Allows farmers to insert in the system data about their production and any problem they face
- *G.6*: Allows farmers to request for help and suggestions by agronomists and other farmers
- *G.7*: Allows farmers to create discussion forums with the other farmers
- *G.8*: Allows agronomists to insert the area they are responsible of
- *G.9*: Allows agronomists to receive information about requests for help
- *G.10*: Allows agronomists to answer to requests for help from farmers
- *G.11*: Allows agronomists to visualize data concerning weather forecasts in the area
- *G.12*: Allows agronomists to visualize data concerning the best performing farmers in the area
- *G.13*: Allows agronomists to visualize a daily plan to visit farms in the area
- *G.14*: Allows agronomists to update a daily plan to visit farms in the area
- *G.15*: Allows agronomists to confirm the execution of the daily plan at the end of each day
- *G.16*: Allows agronomists to specify the deviations from the daily plan at the end of the day

1.2 Scope

The aim of the DREAM software product is to develop and adopt anticipatory governance models for food systems to strengthen data-driven state policy.

It takes care of the acquisition and management of all data collected in order to support the work of farmers, agronomists and policy makers.

The system aims to collect data not only from sensors located throughout the territory, but also from farmers. The analysis of the acquired data aims to improve the production of farmers.

Low-performing farmers are identified by policy makers and supported / helped by the best-performing ones.

Everything is supervised by agronomists who take care of their own geographical areas of competence.

To better understand all the phenomena involved, we distinguish them into two types according to the World and Machine paradigm [M. Jackson and P. Zane]. The World is the environment surrounding the system, while the Machine is the system itself.

1.2.1 World and Shared phenomena

Phenomenon	Who controls it?	Is it shared?
1st phenomena	W	Y
2nd phenomena	W	Y
2nd phenomena	W	Y
2nd phenomena	W	Y
2nd phenomena	W	Y
2nd phenomena	W	Y

1.3 Definitions, Acronyms, Abbreviations

Vuoto

1.3.1 Definitions

Vuoto

1.3.2 Acronyms

Vuoto

1.3.3 Abbreviations

Vuoto

1.4 Revision History

Vuoto

1.5 Reference Documents

Vuoto

1.6 Document Structure

Vuoto

Chapter 2

Overall Description

Vuoto

2.1 Product Perspective

Vuoto

2.1.1 Scenarios

Vuoto

2.1.2 Class Diagrams

Vuoto

2.1.3 State Charts

Vuoto

2.2 Product functions

Vuoto

2.3 Users Characteristics

Vuoto

2.4 Assumptions, dependencies and constraints

Vuoto

2.4.1 Domain Assumptions

Vuoto

2.4.2 Dependencies

Vuoto

Chapter 3

Specific Requirements

Vuoto

3.1 External Interface Requirements

Vuoto

3.1.1 User Interfaces

Vuoto

3.1.2 Hardware Interfaces

Vuoto

3.1.3 Software Interfaces

Vuoto

3.1.4 Communication Interfaces

Vuoto

3.2 Functional Requirements

Vuoto

3.2.1 Use Case Diagrams

Vuoto

3.2.2 Use Case Analysis

Vuoto

3.2.3 Sequence Diagrams

Vuoto

3.2.4 Requirements

Vuoto

3.2.5 Traceability Matrix

Vuoto

3.3 Performance Requirements

Vuoto

3.4 Design Constraints

Vuoto

3.4.1 Standards compliance

Vuoto

3.4.2 Hardware limitations

Vuoto

3.4.3 Any other constraint

Vuoto

3.5 Software System Attributes

Vuoto

3.5.1 Reliability

Vuoto

3.5.2 Availability

Vuoto

3.5.3 Security

Vuoto

3.5.4 Maintainability

Vuoto

3.5.5 Portability

Vuoto

Chapter 4

Formal Analysis using Alloy

Vuoto

4.1 Alloy Model

Vuoto

4.1.1 Signatures

Vuoto

4.1.2 Facts

Vuoto

4.1.3 Assertions

Vuoto

4.1.4 Analysis Results

Vuoto

Chapter 5

Effort Spent

Vuoto

Bibliography

- [1] Elisabetta Di Nitto (2021), *Software Engineering 2 - Lectures*.
- [2] Michael A. Jackson (1995), *The World and the Machine*, <http://mcs.open.ac.uk/mj665/icse17kn.pdf>.
- [3] ETH Zürich (2009), *Requirements Specification*, se.inf.ethz.ch/courses/2011a_spring/soft_arch/exercises/02/Requirements_Specification.pdf.