Module Interface Specification for CRLP (Concrete Remaining Life Perdiction)

Yi-Leng Chen

March 18, 2024

1 Revision History

Date	Version	Notes
March 18	1.0	Initial Documentation

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at GitHub

Contents

3 Introduction

The following document details the Module Interface Specifications for CRLP (Concrete Remaining Life Perdiction)

Complementary documents include the System Requirement Specifications and Module Guide . The full documentation and implementation can be found at GitHub repository

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | ... | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by CRLP.

Data Type	**Notation**	**Description**
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of CRLP uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, CRLP uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**
Hardware-Hiding Module	
Behaviour-Hiding Module	Input Parameters Module Input Format Module Output Format Module Control Module Prediction Calculate Equations Module
Software Decision Module	Sequence Data Structure Module

Table 1: Module Hierarchy

6 MIS of Control Module

The module provides the main program and GUI to call other modules.

6.1 Module

main

6.2 Uses

- Input Parameters Module
- Input Format Module
- Output Format Module
- Prediction Calculate Equations Module

6.3 Syntax

6.3.1 Exported Constants

6.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
main	-	-	-

6.4 Semantics

6.4.1 State Variables

None

6.4.2 Environment Variables

None

6.4.3 Assumptions

None

6.4.4 Access Routine Semantics

main():

- transition: Interact with the GUI and control other modules by following these steps:
  - The necessary data is input by conducting the Input Parameters Module (M2).
  - The Input Format Module (M3) converts the input data into the data structure.
  - Once the verified data is obtained, the Prediction Calculate Equations Module (M6) begins the calculation process.
  - The result is then returned by the Output Format Module (M4).
- output: None
- exception: The exceptions are listed in corresponding submodules.

6.4.5 Local Functions

None

7 MIS of Input Parameters Module

The module stores the parameters necessary for CRLP, including concrete properties and weather conditions.

7.1 Module

parameters

7.2 Uses

- Hardware-Hiding Module

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
parameters	-	-	-

7.4 Semantics

The parameters data structure is utilized to store the input data passed by the Input Format Module.

7.4.1 State Variables

The following theories are retrieved from section 4.1.1 of the VnVPlan document Chen (2024), which aims to identify the symbols used in each equation.

1. Theory 1
   C_{const} (constant chloride concentration at the concrete surface): $\mathbb{N}$
   C_x (chloride concentration at depth x): $\mathbb{R}$
   D_{cl} (chloride ion diffusion coefficient): $\mathbb{R}$
   x (depth): $\mathbb{R}$

2. Theory 2
   L (remaining uncarbonated cover): $\mathbb{R}$
   R_c (rate of carbonation): $\mathbb{R}$

3. Theory 3
   k_a (coefficient of active corrosion): $\mathbb{R}$
   k_c (quality coefficient of concrete): $\mathbb{R}$
   k_e (coefficient of environment): $\mathbb{R}$
   C_R (thickness of concrete cover): $\mathbb{R}$

4. Theory 4
   M_{70} (months that relative humidity below 70%): $\mathbb{Z}$
   M_{100} (months that relative humidity between 70% and 100 %): $\mathbb{Z}$
   M_{rain} (months that rain occurs): $\mathbb{Z}$

5. Theory 5
   A_d (cumulative depth of deterioration after t_{use} years): $\mathbb{N}$
   A_{df} (concrete cover thickness after t_{use} years): $\mathbb{N}$
   k_d (constant encapsulates various factors affecting the degradation of concrete): $\mathbb{N}$
   n (time order): $\mathbb{R}$

7.4.2 Environment Variables

Keyboard: The hardware component used for input. This module assumes the presence of a keyboard for data entry.

7.4.3 Assumptions

None

7.4.4 Access Routine Semantics

parameters():

- transition: This module contains a data structure for storing the input values formatted by the Input Format Module.
- output: None
- exception: None

7.4.5 Local Functions

None

8 MIS of Input Format Module

The module converts the input data into the data structure utilized by the input parameters module.

8.1 Module

input_format

8.2 Uses

- Input Parameters Module

8.3 Syntax

8.3.1 Exported Constants

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
load_params	String, $\mathbb{R}$	$\mathbb{R}$	input_Error,input_SourceError
verify_params	String, $\mathbb{R}$	-	badYear,badMonth,badValue

8.4 Semantics

8.4.1 State Variables

IsSourceExist: Boolean # Ensure that the necessary columns of data exist on the Government of Canada website.

8.4.2 Environment Variables

Data source: When conducting the prediction process using weather data, the script retrieves data from the Government of Canada website.

8.4.3 Assumptions

None

8.4.4 Access Routine Semantics

verify_params():

- transition: The following steps are used to load, verify, and store the input data:
  - Verification of the format of the source website is necessary only for weather data prediction.

-Extract the input columns from the website one by one.
-Verify all inputs, specifically ensuring there are no missing columns.
-Store the inputs in the data structure.

- output: *out*:= None
- exception: exc:=

$A_d < 0 \ \vee A_{df} < 0$ $\Longrightarrow$ badYear
$M_{70} \wedge M_{100} \wedge M_{rain} \neq 12$ $\Longrightarrow$ badMonth
$M_{70} < 0 \vee M_{100} < 0 \vee M_{rain} < 0$ $\Longrightarrow$ badMonth
$\nexists H_r$ (relative humidity) $\Longrightarrow$ input_SourceError

9 MIS of Output Format Module

9.1 Module

output_verify

9.2 Uses

- Hardware Hiding Modules

9.3 Syntax

9.3.1 Exported Constants

9.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
output	String, $\mathbb{R}$	String	input_Error,input_SourceError

9.4 Semantics

9.4.1 State Variables

None

9.4.2 Environment Variables

Screen: The calculation result will be displayed on the computer screen.

9.4.3 Assumptions

None

9.4.4 Access Routine Semantics

output():

- transition: This module will output the result to the user's device.
- output: *out*:= String
- exception: exc:= None

10 MIS of Prediction Calculate Equations Module

10.1 Module

calculation

10.2 Uses

- Input Parameters Module

10.3 Syntax

10.3.1 Exported Constants

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
calculate	C_{const}: $\mathbb{N}$, C_x: $\mathbb{R}$, D_{cl}: $\mathbb{R}$, x: $\mathbb{R}$, L: $\mathbb{R}$, R_c: $\mathbb{R}$, k_a: $\mathbb{R}$, k_c: $\mathbb{R}$, k_e: $\mathbb{R}$, C_R: $\mathbb{R}$, M_{70}: $\mathbb{Z}$, M_{100}: $\mathbb{Z}$, M_{rain}: $\mathbb{Z}$, A_d: $\mathbb{N}$, A_{df}: $\mathbb{N}$, k_d: $\mathbb{N}$, n: $\mathbb{R}$	String	MissingDataError

10.4 Semantics

10.4.1 State Variables

None

10.4.2 Environment Variables

None

10.4.3 Assumptions

None

10.4.4 Access Routine Semantics

calculate(params):

- transition: After formatting the input data, this module will perform the calculation process.
- output: The remaining service life of specific concrete structure.
  out:= t_{remain} (remaining service life of concrete)
- exception: exc:= missing any input parameters $\Longrightarrow$ MissingDataError

References

Yi-Leng Chen. Verification and validation plan for concrete remaining life prediction(crlp). https://github.com/kypss94132/CAS741_YiLeng-Chen/blob/main/docs/VnVPlan/VnVPlan.pdf, 2024.

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995.