A Methodology for Building a Generic Irrigation Model

1 Introduction

The aim of this document is to present a methodology for building a generic irrigation model. The generic model would serve both the developers/designers of an irrigation expert system and the implementers of an expert system tool. The primary goal for developing this methodology, and consequently a tool based on it, is to facilitate the rapid development of an irrigation component by offering the system builder a template that can be easily filled To do so, we have aimed to identify and capture all knowledge that is related to the irrigation task, regardless of the crop, and identify concepts that vary from one crop to another. A typical irrigation expert system is made out of concepts and relations, on top of which a task layer is built. By identifying concepts that are common across any irrigation system, we can enhance the re-use of these concepts. We define those as static concepts. Dynamic concepts are those, which may vary from one crop to another. Though we can not accurately predict all of those, we can predict concepts to which they are related (plant, plantation type, etc) . We can consequently identify the relations that these concepts will affect and enable those to be edited to modify an irrigation model.

Irrigation is part of the production phase of any crop, and though the irrigation requirements for each crop may vary, a number of basic concepts and irrigation determination techniques are shared among all. It is the purpose of this document to outline all these similarities in order to facilitate the process of building any irrigation sub-system.

In the design presented, an irrigation schedule could be obtained either on daily or weekly bases. The first depends on the availability of weather station for providing the daily climate input parameters. Where, the second depends on the availability of climate reference data. The irrigation calculations cover the case of planting within an open field, a high tunnel, or a low tunnel. Therefore, the two methods of Et0 calculations (Hargerve, Pennman) have been considered since the first method applies only to high tunnels, while the second is valid for the other two farming types. The irrigation model included, also caters for flooding, or drip irrigation systems.

It is important to note that this document is mostly based on an earlier technical report which is: “Generic Irrigation design Applied in tomato Crop” TR/CLAES/72/99.5

2 Built in/Required Concepts

2.1 General (General Ontology)

2.1.1 Task Parameters

Name: task_parameters
Properties:

Name	current_month

Description	A number denoting the current month
Source of Value	System Derived
Type	Integer
Legal values	Number-range(1,12)

Name	**next_month**
Description	A number denoting the next month
Source of Value	Derived
Type	Integer
Legal values	Number-range(1,12)

Name	**current_day**
Description	A number denoting the current day
Source of Value	System Derived
Type	Integer
Legal values	Number-range(1,31)

Name	**current_date**
Description	A date denoting the current date
Source of Value	System Derived
Type	Date
Legal values	Date

Name	**week_number**
Description	A number denoting ??
Source of Value	System Derived
Type	Integer
Legal values	Number-range(1,53)

Name	**age**
Description	A number denoting the age of the plant (in days?)
Source of Value	System Derived
Type	Integer
Legal values	Number-range(1,1000)

Name	**previous_accumulated_eta**
Description	accumulated eta
Source of Value	Unkown?
Type	Integer
Legal values	Number-range(0,1000)

Name	**eta_plus_eta_acc**

Description	Current eta + accumulated eta
Source of Value	Derived from function: eta_plus_eta_acc_f (page 71)
Type	Integer
Legal values	Number-range(0,1000)

Name	**pawc_decrease_twenty_per**
Description	The value of pawc reduced by 20%
Source of Value	Derived from function: pawc_decrease_twenty_per_f (page 71)
Type	Integer
Legal values	Number-range(0,1000)

Name	**interval_number**
Description	
Source of Value	Unkown?
Type	Integer
Legal values	Number-range(0,1000)

2.2 Plantation Related (*Plantation Ontology)*

2.2.1 Soil

Name: Soil
Properties:

Name	**Texture**
Description	Texture of the soil
Source of Value	DB
Type	Nominal /Single
Legal values	'clay', 'clay loam', 'loamy sand', 'sand,' 'sandy clay', 'sandy clay loam', 'sandy loam', 'silt loam', 'silty clay', 'silty clay loam', 'silty loam'

Name	**Type**
Description	Denotes the classification of the soil texture
Source of Value	Relation: Soil_type (page 46)
Type	Nominal /Single
Legal values	fine, medium, coarse

Name	**Sp**
Description	soil saturation percentage
Source of Value	DB or Table: SBD_table (page 58)
Type	Real
Legal values	Number-range(0,1000)

Name	**Sbd**
Description	soil bulk density
Source of Value	DB or Table: SP_table (page 58)
Type	Real

Name	Sbd
Legal values	Number-range(0,1000)

Name	Ec
Description	soil salinity
Source of Value	DB
Type	Real
Legal values	Number-range(0,6)

2.2.2 Water

Name: water
Properties:

Name	eciw
Description	Electronic conductivity of irrigation water (Ds/m)
Source of Value	DB
Type	Real
Legal values	Number-range(0,3)

2.2.3 Climate

Name: climate
Properties:

Name	avg_tc
Description	Average temperature
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(0,40)

Name	avg_rh
Description	Average relative humidity
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(0,100)

Name	ash
Description	Actual sun shine hours
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(5,17)

Name	msh
Description	Maximum sun shine hours
Source of Value	DB or Function: msh_f
Type	Real
Legal values	Number-range(5,17)

Name	ra
Description	Extra radiation
Source of Value	DB or Function: ra_f (page 62)
Type	Real
Legal values	Number-range(3,17)

Name	max_rh
Description	Maximum relative humidity
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(0,100)

Name	mws
Description	Mean wind speed
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(0,100)

Name	dws
Description	Day wind speed
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(0,100)

Name	nws
Description	Night wind speed
Source of Value	DB. The value is retrieved based on a month number
Type	Real
Legal values	Number-range(0,100)

Name	msh_par_a
Description	Maximum sunshine hours parameter (a)
Source of Value	Table: msh_a_table
Type	Real
Legal values	Number-range(-10,17)

Name	msh_par_b
Description	Maximum sunshine hours parameter (b)
Source of Value	Table: msh_b_table
Type	Real
Legal values	Number-range(-10,17)

Name	ra_par_a

Description	Solar radiation parameter (a)
Source of Value	Table: ra_a_table
Type	Real
Legal values	Number-range(-10,17)

Name	**ra_par_b**
Description	Solar radiation parameter (a)
Source of Value	Table: ra_b_table
Type	Real
Legal values	Number-range(-10,17)

Name	**avg_tc_next_month**
Description	Average temperature for next month
Source of Value	DB
Type	Real
Legal values	Number-range(0,40)

Name	**avg_rh_next_month**
Description	Average relative humidity for next month
Source of Value	DB
Type	Real
Legal values	Number-range(0,100)

Name	**ash_next_month**
Description	Actual sun shine hours for next month
Source of Value	DB
Type	Real
Legal values	Number-range(5,17)

Name	**msh_next_month**
Description	Maximum sun shine hours for next month
Source of Value	DB or Function: msh_f (page 62)
Type	Real
Legal values	Number-range(5,17)

Name	**ra_next_month**
Description	Extra radiation for next month
Source of Value	DB or Function: ra_f (page 62)
Type	Real
Legal values	Number-range(3,17)

2.2.4 Plant

Provided here, are some build in properties for the concept plant. However, the knowledge engineer should add any other concepts that are appropriate to the kind of plant being addressed.

Properties:

Name	**init_stage**
Description	The length of the initiation stage in days
Source of Value	User or from ***relation***: growth_stages (page 46)
Type	Integer
Legal values	Number-range(0,1000)

Name	**ve_stage**
Description	The length of the vegetative stage in days
Source of Value	User or from ***relation***: growth_stages (page 46)
Type	Integer
Legal values	Number-range(0,1000)

Name	**fl_stage**
Description	The length of the flowering stage in days
Source of Value	User or from ***relation***: growth_stages (page 46)
Type	Integer
Legal values	Number-range(0,1000)

Name	**fr_stage**
Description	The length of the fruiting stage in days
Source of Value	User or from ***relation***: growth_stages (page 46)
Type	Integer
Legal values	Number-range(0,1000)

Name	**growth_period**
Description	The number of days needed for a plant to reach its growth.
Source of Value	***Relation***: growth_period_f (page 63)
Type	Integer
Legal values	Number-range(0,1000)

Name	**init_ve_stage**
Description	The number of days after which both the initiation and vegetative stage would be complete
Source of Value	***Function***: init_ve_stage_f (page 63)
Type	Integer
Legal values	Number-range(0,1000)

Name	**init_ve_fl_stage**
Description	Initiation, vegetative and flowering stage
Source of Value	***Function:*** init_ve_fl_stage_f (page 63)

Type	Integer
Legal values	Number-range(0,1000)

2.2.5 Current_Planting

Properties:

Name	**no_of_plants**
Description	Number of plants
Source of Value	From DB
Type	Integer
Legal values	Number-range(021000)

Name	**date**
Description	??
Source of Value	From DB
Type	Date
Legal values	A valid date

Name	**agriculture_method**
Description	Method of cultivation
Source of Value	DB
Type	Nominal
Legal values	Seedling, Transplanting

Name	**death_of_plants**
Description	A flag that contains info about whether plants are dying
Source of Value	User
Type	Nominal
Legal values	Yes, No

2.2.6 Farm

Name: Farm
Properties:

Name	**id**
Description	A name to identify the farm
Source of Value	User
Type	String
Legal values	

Name	**latitude**
Description	Farm latitude
Source of Value	DB
Type	Real
Legal values	Number-range(0,1000)

Name	**altitude**
Description	Farm altitude

Source of Value	DB
Type	Real
Legal values	Number-range(0,1000)

Name	**area**
Description	Farm area
Source of Value	DB
Type	Real
Legal values	Number-range(0,20000)

Name	**drainage_system**
Description	Farm's drainage_system
Source of Value	DB
Type	Nominal/Single
Legal values	Good, Medium, Bad

Name	**planting_type**
Description	Farm's planting environment
Source of Value	DB
Type	Nominal/Single **(Necessary)**
Legal values	Open_field, Low_tunnel, High_tunnel

2.2.7 low_tunnel&open_field

Name: low_tunnel&open_field
Sub-type of: farm
Properties:

Name	**dist_b_plants**
Description	Distance between plants
Source of Value	DB
Type	Real **(Necessary)**
Legal values	Number-range(0,1000)

Name	**dist_b_rows**
Description	Distance between plant rows
Source of Value	DB
Type	Real **(Necessary)**
Legal values	Number-range(0,1000)

2.2.8 Low_tunnel

Name: Low_tunnel
Sub-type of: low_tunnel&open_field

2.2.9 Open_field

Name: Open_field
Sub-type of: : low_tunnel&open_field

2.2.10 high_tunnel

Name: low_tunnel&open_field
Sub-type of: farm
Properties:

Name	length
Description	The length of the tunnel
Source of Value	DB
Type	Real
Legal values	Number-range(0, 4000)

Name	width
Description	The width of the tunnel
Source of Value	DB
Type	Real
Legal values	Number-range(0, 4000)

Name	Plastic_age
Description	The age of the plastic used
Source of Value	DB
Type	Real
Legal values	Number-range(1, 5)

2.3 Operations Related (*Operations Ontology)*

2.3.1 Operations

Name: operations
Properties:

Name	done
Description	A flag
Source of Value	????
Type	Nominal /Single
Legal values	'yes', 'no'

2.3.2 high_tunnel_operations

Name: high_tunnel_operations
Sub-type of: operations

2.3.3 mulch

Name: mulch
Sub-type of: high_tunnel_operations
Properties:

Name	done
Description	A flag
Source of Value	DB
Type	Nominal /Single
Legal values	'yes', 'no'

2.3.4 tunnel_painting

Name: tunnel_painting
Sub-type of: high_tunnel_operations
Properties:

Name	done
Description	A flag
Source of Value	DB
Type	Nominal /Single
Legal values	'yes', 'no'

Name	tunnel_painting_factor
Description	
Source of Value	***Relation***: tunnel_painting_factor_r (page 46)
Type	Real
Legal values	Number-range(0,100)

2.3.5 soil_sterilization

Name: soil_sterilization
Sub-type of: high_tunnel_operations
Properties:

Name	method
Description	The method of soil sterilization
Source of Value	DB
Type	Nominal /Single
Legal values	'mythyle bromide', 'bazamide', 'solarization', 'others', 'none'

2.3.6 irrigation

Sub-type of: operations
Properties:

Name	method
Description	The irrigation method
Source of Value	DB
Type	Nominal /Single
Legal values	'drip', 'flooding'

Name	schedule_type
Description	The required type of irrigation schedule
Source of Value	User
Type	Nominal /Single
Legal values	'daily', 'weekly'

Name	irrigation_efficiency
Description	The irrigation efficiency
Source of Value	***Table:*** irrigation_efficiency_t (page 61)
Type	Integer
Legal values	Number-range(0,100)

Name	number_of_drippers
Description	The existing number of drippers
Source of Value	DB
Type	Integer
Legal values	Number-range(100, 4000)

Name	rate_of_dripper_flow
Description	The rate of a dripper's flow
Source of Value	DB
Type	Real **(Necessary)**
Legal values	Number-range(1,100)

Name	control_of_dripper
Description	????
Source of Value	DB
Type	Nominal /Single
Legal values	'yes', 'no'

Name	controled_water
Description	????
Source of Value	DB
Type	Nominal /Single **(Necessary)**
Legal values	'yes', 'no'

Name	user_suggested_interval
Description	????
Source of Value	DB
Type	Integer
Legal values	Number-range(1, 30)

Name	saa
Description	The soil absorbed area
Source of Value	***Table:*** Saa_t (page 61)
Type	Integer
Legal values	Number-range(0,100)

Name	last_irr_date
Description	The date on which the last irrigation was performed
Source of Value	User
Type	Date
Legal values	

Name	Last_wr
Description	last water requirement (what does that mean?)
Source of Value	User
Type	Real
Legal values	Number-range(0,1000)

2.4 Material Related (*Material Ontology)*

This ontology is to be specified by the knowledge engineer depending on the crop being addressed.

2.5 Plantation Factors Ontology

This ontology is to be specified by the knowledge engineer depending on the crop being addressed.

2.5.1 plantation_factors

The value of this concept is required and should be calculated depending on the crop being planted. Further clarification of what plantation_factors.value is should be given and included in this document.

Properties:

Name	**value**
Description	The value of plantation_factors.
Source of Value	Depends on the crop.
Type	Real
Legal values	Number-range(0, 1000)

2.6 Computational Ontology: Et0 Ontology

2.6.1 et0

et0 is a very important factor for the irrigation task. However, the way it's value is highly dependent on whether planting is done within an open field, or a high or low tunnel. When planting is performed within a high tunnel, this value is calculated using what is known as the Hargerve method. If When planting is performed within a low tunnel or in an open field, then the Pennman method is used. Both the Hargerve and Pennman methods, depend on other factors in order to correctly derive correct values. Because of these dependencies, a dependency graph is used for the derivation of the value of et0. This graph is given in put in appendix or something :

Properties:

Name	**value**
Description	The value of et0 which is the potential evotranspiration.
Source of Value	Derived from a dependency graph. The first node in this graph is the pre-condition relation: et0_pcf (page 47)
Type	Real
Legal values	Number-range(0, 1000)

2.6.2 et0_hargerve

Name: et0_hargerve
Sub-type of: et0
Properties:

Name	**use**
Description	A flag to indicate whether the hargerve method should be used to calculate the value of et0 or not
Source of Value	Derived from relation: et0_pcf (page 47)
Type	Nominal /Single

Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of et0_hargerve which is the potential evotranspiration calculated by the hargerve method.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: et0_hargerve_pcf (page 47)
Type	Real
Legal values	Number-range(0, 1000)

2.6.3 et0_pennman

Name: et0_ pennman
Sub-type of: et0
Properties:

Name	**use**
Description	A flag to indicate whether the pennman method should be used to calculate the value of et0 or not
Source of Value	Derived from relation: et0_pcf (page 47)
Type	Nominal /Single
Legal values	'yes', 'no' (default = 'no')

Name	**value**
Description	The value of et0_ pennman which is the potential evotranspiration calculated by the pennman method
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: et0_pennman _pcf (page 47)
Type	Real
Legal values	Number-range(0, 1000)

2.6.4 et0_smooth_hargerve

Name: et0_ smooth _hargerve
Sub-type of: et0_ hargerve
Properties:

Name	**use**
Description	A flag to indicate whether or not to calculate the value of et0_ smooth _hargerve
Source of Value	Derived from relation et0_hargerve_pcf (page 47)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of et0_ smooth _hargerve.
Source of Value	Derived from a dependency graph. The node used to derive this value is the function: et0_smooth_hargerve_f (page 63)
Type	Real
Legal values	Number-range(0, 1000)

2.6.5 et0_har_c

Name: et0_har_c
Sub-type of: et0_hargerve
Properties:

Name	use
Description	A flag to indicate whether the value of et0_har_c should be calculated or not
Source of Value	Derived from relation: et0_hargerve_pcf (page 47)
Type	Nominal /Single
Legal values	'yes', 'no' (default = 'no')

Name	value
Description	The value of et0_har_c which is the potential evotranspiration for the current month calculated by the hargerve method
Source of Value	Derived from a dependency graph. The node used to derive this value is the function: et0_har_c_f (page 63)
Type	Real
Legal values	Number-range(0, 1000)

2.6.6 et0_har_n

Name: et0_har_n
Sub-type of: et0_hargerve
Properties:

Name	use
Description	A flag to indicate whether the value of et0_har_n should be calculated or not
Source of Value	Derived from relation: et0_hargerve_pcf (page 47)
Type	Nominal /Single
Legal values	'yes', 'no' (default = 'no')

Name	value
Description	The value of et0_har_n which is the potential evotranspiration for the next month calculated by the hargerve method
Source of Value	Derived from a dependency graph. The node used to derive this value is the function: et0_har_n_f (page 64)
Type	Real
Legal values	Number-range(0, 1000)

2.6.7 control_f

Name: control_f
Sub-type of: et0_smooth_hargerve
Properties:

Name	value
Description	The value of the control factor control_f.
Source of Value	Derived from a dependency graph. The node used to derive this value is the relation: control_f_r (page 48)
Type	Real

Legal values	Number-range(0, 1000)

2.6.8 l_rh_c

Name: l_rh_c
Sub-type of: et0_ penman
Properties

Name	Use
Description	A flag to indicate whether the value of l_rh_c should be calculated or not
Source of Value	Derived from relation: et0_pennman_pcf (page 47)
Type	Nominal /Single
Legal values	'yes', 'no' (default = 'no')

Name	Value
Description	The value of l_rh_c which is the low maximum relative humidity for the current month
Source of Value	Derived from a dependency graph. The node used to derive this value is function of l_rh_c_f (page 64)
Type	Real
Legal values	Number-range(0, 1000)

2.6.9 m_rh_c

Name: m_rh_c
Sub-type of: et0_ penman
Properties

Name	Use
Description	A flag to indicate whether the value of m_rh_c should be calculated or not
Source of Value	Derived from relation: et0_pennman_pcf (page 47)
Type	Nominal /Single
Legal values	'yes', 'no' (default = 'no')

Name	Value
Description	The value of m_rh_c which is the medium maximum relative humidity for the current month
Source of Value	Derived from a dependency graph. The node used to derive this value is function of m_rh_c_f (page 64)
Type	Real
Legal values	Number-range(0, 1000)

2.6.10 h_rh_c

Name: h_rh_c
Sub-type of: et0_ penman
Properties

Name	Use
Description	A flag to indicate whether the value of h_rh_c should be calculated or not
Source of Value	Derived from relation: et0_pennman_pcf (page 47)

Type	Nominal /Single
Legal values	‘yes’, ‘no’ (default = ‘no’)

Name	**Value**
Description	The value of m_rh_c which is the high maximum relative humidity for the current month
Source of Value	Derived from a dependency graph. The node used to derive this value is function of h_rh_c_f (page)
Type	Real
Legal values	Number-range(0, 1000)

2.7 Computational Ontology: Eta Ontology

2.7.1 eta

Name: eta
Properties:

Name	**value**
Description	The value of eta which is the exhaustion of actual water.
Source of Value	Derived from function: eta_f (page 65)
Type	Real
Legal values	Number-range(0, 1000)

2.7.2 gc

Name: gc
Sub-type of: eta
Properties:

Name	**use**
Description	A flag used to indicate whether this node should be invoked. Something seems wrong in the given graph as this seems to be a DG on its own
Source of Value	Unknown
Type	Nominal /Single
Legal values	‘yes’, ‘no’ no’ (default = ‘yes’) ? It seems like it yes by default?

Name	**value**
Description	The value of gc, which is actually the green cover area.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: gc_pcf (page 48)
Type	Real
Legal values	Number-range(0, 1000)

2.7.3 gc_init_st

Name: gc_init_st
Sub-type of: gc
Properties:

Name	**use**
Description	A flag used to indicate whether or not to calculate the green cover area for the initiation stage.

Source of Value	Derived from relation: gc_pcf (page 48)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The green cover area for the initiation stage.
Source of Value	Derived from function: gc_init_st_f (page 66)
Type	Real
Legal values	Number-range(0, 1000)

2.7.4 gc_ve_st

Name: gc_ve_st
Sub-type of: gc
Properties:

Name	**use**
Description	A flag used to indicate whether or not to calculate the green cover area for the vegetative stage.
Source of Value	Derived from relation: gc_pcf (page 48)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The green cover area for the vegetative stage.
Source of Value	Derived from function: gc_ve_st_f (page 66)
Type	Real
Legal values	Number-range(0, 1000)

2.7.5 gc_fl_st

Name: gc_fl_st
Sub-type of: gc
Properties:

Name	**use**
Description	A flag used to indicate whether or not to calculate the green cover area for the flowering stage.
Source of Value	Derived from relation: gc_pcf (page 48)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The green cover area for the flowering stage.
Source of Value	Derived from function: gc_fl_st_f (page 66)
Type	Real
Legal values	Number-range(0, 1000)

2.7.6 gc_ fr _st

Name: gc_ fr _st
Sub-type of: gc
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the green cover area for the fruiting stage.
Source of Value	Derived from relation: gc_pcf (page 48)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The green cover area for the fruiting stage.
Source of Value	Derived from function: gc_ fr _st_f (page 66)
Type	Real
Legal values	Number-range(0, 1000)

2.7.7 kc

Name: kc
Sub-type of: eta
Properties:

Name	use
Description	A flag used to indicate whether this node should be invoked. Something seems wrong in the given graph as this seems to be a DG on its own
Source of Value	Unknown
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'yes') ? It seems like it yes by default?

Name	value
Description	The value of kc, which is a crop coefficient.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: kc_pcf (page 49)
Type	Real
Legal values	Number-range(0, 1000)

2.7.8 kc_init_st

Name: kc_init_st
Sub-type of: kc
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient area for the initiation stage.
Source of Value	Derived from relation: kc_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient for the initiation stage.
Source of Value	Derived from function: kc_init_st_f (page 66)
Type	Real
Legal values	Number-range(0, 1000)

2.7.9 kc_ve_st

Name: kc_ve_st
Sub-type of: kc
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the vegetative stage.
Source of Value	Derived from relation: kc_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient for the vegetative stage.
Source of Value	Derived from function: kc_ve_st_f (page 66)
Type	Real
Legal values	Number-range(0, 1000)

2.7.10 kc_fl_st

Name: kc_fl_st
Sub-type of: kc

Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the flowering stage.
Source of Value	Derived from relation: kc_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient for the flowering stage.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: kc_fl_st_pcf (page 49)
Type	Real
Legal values	Number-range(0, 1000)

2.7.11 kc_ fl_h_and_lt

Name: kc_ fl_h_and_lt
Sub-type of: kc_ fl_st

Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the flowering stage in high and low tunnels
Source of Value	Derived from relation: kc_fl_st_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient kc for the flowering stage in high and low tunnels
Source of Value	Derived from function: kc_fl_h_and_lt_f (page 67)
Type	Real
Legal values	Number-range(0, 1000)

2.7.12 kc_ fl_of

Name: kc_ fl_h_and_lt
Sub-type of: kc_ fl_st

Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the flowering stage in open fields
Source of Value	Derived from relation: kc_fl_st_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient kc for the flowering stage in open fields
Source of Value	Derived from function: kc_fl_of_f (page 67)
Type	Real
Legal values	Number-range(0, 1000)

2.7.13 kc_ fr _st

Name: kc_ fr _st
Sub-type of: kc
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the fruiting stage.
Source of Value	Derived from relation: kc_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' (default = 'no')

Name	value
Description	The crop coefficient for the fruiting stage.
Source of Value	Derived from a dependency graph. The node used to derive this

	value is the pre-condition relation: kc_fr_st_pcf (page 49)
Type	Real
Legal values	Number-range(0, 1000)

2.7.14 kc_ fr_h_and_lt

Name: kc_ fr_h_and_lt
Sub-type of: kc_ fr_st

Properties:

Name	**use**
Description	A flag used to indicate whether or not to calculate the crop coefficient for the fruiting stage in high and low tunnels
Source of Value	Derived from relation: kc_fr_st_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The crop coefficient kc for the fruiting stage in high and low tunnels
Source of Value	Derived from function: kc_fr_h_and_lt_f (page 67)
Type	Real
Legal values	Number-range(0, 1000)

2.7.15 kc_ fr_of

Name: kc_ fr_h_and_lt
Sub-type of: kc_ fr_st

Properties:

Name	**use**
Description	A flag used to indicate whether or not to calculate the crop coefficient for the fruiting stage in open fields
Source of Value	Derived from relation: kc_fr_st_pcf (page 49)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The crop coefficient kc for the fruiting stage in open fields
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: kc_fr_of_pcf (page 50)
Type	Real
Legal values	Number-range(0, 1000)

2.7.16 kc_ fr_l_age

Name: kc_ fr_l_age
Sub-type of: kc_ fr_of

Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the fruiting stage in open fields when the plant less than a given age
Source of Value	Derived from relation: kc_fr_of_pcf (page 50)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient kc for the fruiting stage in open fields when the plant is less than a given age
Source of Value	Derived from function: kc_fr_l_160_f (page 67)
Type	Real
Legal values	Number-range(0, 1000)

2.7.17 kc_ fr_g_age

Name: kc_ fr_g_age
Sub-type of: kc_ fr_of

Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the crop coefficient for the fruiting stage in open fields when the plant is more than a given age
Source of Value	Derived from relation: kc_fr_of_pcf (page 50)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The crop coefficient kc for the fruiting stage in open fields when the plant is more than a given age
Source of Value	Derived from function: kc_fr_g_160_f (page 67)
Type	Real
Legal values	Number-range(0, 1000)

2.7.18 depression_factor

Name: depression_factor
Sub-type of: eta
Properties:

Name	use
Description	A flag used to indicate whether this node should be invoked.
Source of Value	Unknown
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'yes') ? It seems like it yes by default?

Name	value

Description	The value of the depression factor.
Source of Value	Derived from relation: depression_factor_r (page 50)
Type	Real
Legal values	Number-range(0, 1000)

2.7.19 unit_factor

Name: unit_factor
Sub-type of: eta
Properties:

Name	**use**
Description	A flag used to indicate whether this node should be invoked.
Source of Value	Unknown
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'yes') ? It seems like it yes by default?

Name	**value**
Description	The value of the unit factor.
Source of Value	Derived from relation: unit_factor_r (page 50)
Type	Real
Legal values	Number-range(0, 1000)

2.8 Computational Ontology: Pawc Ontology

2.8.1 Pawc

Name: pawc
Properties:

Name	**value**
Description	The value of ***pawc***, which is the water available in soil
Source of Value	Derived from function: pawc_f (page 68)
Type	Real
Legal values	Number-range(0, 1000)

2.8.2 ad

Name: ad
Sub-type of: pawc
Properties:

Name	**value**
Description	The value of ***ad***, which the is the crop allowable water depletion
Source of Value	Derived from table ad_t (page 62)
Type	Real
Legal values	Number-range(0, 1000)

2.8.3 rd

Name: rd
Sub-type of: pawc
Properties:

Name	value
Description	The value of ***rd***, which is root depth
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: rd_pcf (page 51)
Type	Real
Legal values	Number-range(0, 1000)

2.8.4 rd_init_st

Name: rd_init_st
Sub-type of: rd
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the root depth for the initiation stage.
Source of Value	Derived from relation: rd_pcf (page 51)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The root depth for the initiation stage.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: rd_init_st_pcf (page 51)
Type	Real
Legal values	Number-range(0, 1000)

2.8.5 rd_ve_st

Name: rd_ve_st
Sub-type of: rd
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the root depth for the vegetative stage.
Source of Value	Derived from relation: rd_pcf (page 51)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The root depth for the vegetative stage.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: rd_ve_st_pcf (page 52)
Type	Real
Legal values	Number-range(0, 1000)

2.8.6 rd_fl_st

Name: rd_fl_st

Sub-type of: rd
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the root depth for the flowering stage.
Source of Value	Derived from relation: rd_pcf (page 51)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The root depth for the flowering stage.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: rd_fl_st_pcf (page 52)
Type	Real
Legal values	Number-range(0, 1000)

2.8.7 rd_fr_st

Name: rd_fr_st
Sub-type of: rd
Properties:

Name	use
Description	A flag used to indicate whether or not to calculate the root depth for the fruiting stage.
Source of Value	Derived from relation: rd_pcf (page 51)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The root depth for the fruiting stage.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: rd_fr_st_pcf (page 52)
Type	Real
Legal values	Number-range(0, 1000)

2.8.8 rd_init_st_d

Name: rd_init_st_d
Sub-type of: rd_init_st
Properties:

Name	use
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_init_st_pcf (page 51)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of root depth for the initiation stage.

Source of Value	Derived from function: rd_init_st_d_f (page 68)
Type	Real
Legal values	Number-range(0, 1000)

2.8.9 rd_init_st_f

Name: rd_init_st_f
Sub-type of: rd_init_st
Properties:

Name	**use**
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_init_st_pcf (page 51)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of root depth for the initiation stage.
Source of Value	Derived from function: rd_init_st_f_f (page 68)
Type	Real
Legal values	Number-range(0, 1000)

2.8.10 rd_ve_st_d

Name: rd_ve_st_d
Sub-type of: rd_ve_st
Properties:

Name	**use**
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_ve_st_pcf (page 52)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of root depth in the vegetative stage.
Source of Value	Derived from function: rd_ve_st_d_f (page 68)
Type	Real
Legal values	Number-range(0, 1000)

2.8.11 rd_ve_st_f

Name: rd_ve_st_f
Sub-type of: rd_ve_st
Properties:

Name	**use**
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_ve_st_pcf (page 52)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**

Description	The value of root depth in the vegetative stage.
Source of Value	Derived from function: rd_ve_st_f_f (page 68)
Type	Real
Legal values	Number-range(0, 1000)

2.8.12 rd_fl_st_d

Name: rd_fl_st_d
Sub-type of: rd_fl_st
Properties:

Name	**use**
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_fl_st_pcf (page 52)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of root depth in the flowering stage.
Source of Value	Derived from function: rd_fl_st_d_f (page 69)
Type	Real
Legal values	Number-range(0, 1000)

2.8.13 rd_fl_st_f

Name: rd_fl_st_f
Sub-type of: rd_fl_st
Properties:

Name	**use**
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_fl_st_pcf (page 52)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of root depth in the flowering stage.
Source of Value	Derived from function: rd_fl_st_f_f (page 69)
Type	Real
Legal values	Number-range(0, 1000)

2.8.14 rd_fr_st_d

Name: rd_fr_st_d
Sub-type of: rd_fr_st
Properties:

Name	**use**
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_fr_st_pcf (page 52)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of root depth in the fruiting stage.
Source of Value	Derived from function: rd_fr_st_d_f (page 69)
Type	Real
Legal values	Number-range(0, 1000)

2.8.15 rd_fr_st_f

Name: rd_fr_st_f
Sub-type of: rd_fr_st
Properties:

Name	use
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: rd_fr_st_pcf (page 52)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of root depth in the fruiting stage.
Source of Value	Derived from function: rd_fr_st_f_f (page 69)
Type	Real
Legal values	Number-range(0, 1000)

2.8.16 max_rd

Name: max_rd
Sub-type of: rd_fr_st_, rd_init_st_d, rd_init_st_f, rd_ve_st_d, rd_ve_st_f, rd_fl_st_d, rd_fl_st_f, rd_fr_st_d , rd_fr_st_f

Properties:

Name	value
Description	The value of the maximum rooting depth
Source of Value	Derived from table: max_rd_t(page 69)
Type	Real
Legal values	Number-range(0, 1000)

2.8.17 sp_factor

Name: rd
Sub-type of: pawc
Properties:

Name	value
Description	The value of ***sp_factor***, which is soil saturated percentage
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: sp_factor_pcf (page 53)
Type	Real
Legal values	Number-range(0, 1000)

2.8.18 sp_l

Name: sp_l
Sub-type of: sp_factor
Properties:

Name	use
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: sp_factor_pcf (page 53)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of soil saturated percentage.
Source of Value	Derived from function: sp_l_f (page 70)
Type	Real
Legal values	Number-range(0, 1000)

2.8.19 sp_g

Name: sp_g
Sub-type of: sp_factor
Properties:

Name	use
Description	A flag used to indicate whether or not to use this node
Source of Value	Derived from relation: sp_factor_pcf (page 53)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of soil saturated percentage.
Source of Value	Derived from function: sp_g_f (page 70)
Type	Real
Legal values	Number-range(0, 1000)

2.9 Computational Ontology: Interval Ontology

2.9.1 interval

Name: interval
Properties:

Name	value
Description	The value of ***interval***
Source of Value	Derived from a dependency graph. The first node in this graph is the pre-condition relation: interval_pcf (page 53)
Type	Real
Legal values	Number-range(0, 1000), default =1

2.9.2 weekly_basis_interval

Name: weekly_basis_interval
Sub-type of: interval
Properties:

Name	use
Description	A flag to indicate whether to calculate a weekly basis interval
Source of Value	Derived from relation: interval_pcf (page 53)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of weekly_basis_interval.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: weekly_basis_interval_pcf (page 53)
Type	Real
Legal values	Number-range(0, 1000)

2.9.3 interval _weekly

Name: interval_weekly
Sub-type of: weekly_basis_interval
Properties:

Name	use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: weekly_basis_interval_pcf (page 53)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of the weekly irrigation interval.
Source of Value	Derived from function: interval_weekly_f (page 70)
Type	Real
Legal values	Number-range(0, 1000)

2.9.4 user_suggested_interval_weekly

Name: user_suggested_interval_weekly
Sub-type of: weekly_basis_interval
Properties:

Name	use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: weekly_basis_interval_pcf (page 53)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of the weekly irrigation interval.
Source of Value	Derived from function: user_suggested_interval_weekly_f (page

	70)
Type	Real
Legal values	Number-range(0, 1000)

2.9.5 daily_basis_interval

Name: daily_basis_interval
Sub-type of: interval
Properties:

Name	**use**
Description	A flag to indicate whether to calculate a daily basis interval
Source of Value	Derived from relation: interval_pcf (page 53)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of daily _basis_interval.
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: daily_basis_interval_pcf (page 54)
Type	Real
Legal values	Number-range(0, 1000)

2.9.6 irrigation_based_on_eta

Name: irrigation_based_on_eta
Sub-type of: daily_basis_interval
Properties:

Name	**use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: daily_basis_interval_pcf (page 54)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: irrigation_based_on_eta_pcf (page 54)
Type	Real
Legal values	Number-range(0, 1000)

2.9.7 irrigation_based_on_eta

Name: irrigation_based_on_eta
Sub-type of: daily_basis_interval
Properties:

Name	**use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: daily_basis_interval_pcf (page 54)

Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: irrigation_based_on_eta_pcf (page 54)
Type	Real
Legal values	Number-range(0, 1000)

2.9.8 suggest_ daily_interval

Name: suggest_ daily_interval
Sub-type of: daily_basis_interval
Properties:

Name	**use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: daily_basis_interval_pcf (page 54)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: suggest_daily_interval_pcf (page 55)
Type	Real
Legal values	Number-range(0, 1000)

2.9.9 there_is_irrigation_today

Name: there_is_irrigation_today
Sub-type of: irrigation_based_on_eta, daily_basis_interval, suggest_ daily_interval
Properties:

Name	**use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: suggest_daily_interval_pcf (page 55))
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**value**
Description	The value of the weekly irrigation interval.
Source of Value	Derived from function: there_is_irrigation_today_f (page 70)
Type	Real
Legal values	Number-range(0, 1000)

2.9.10 there_is_no_irrigation_today

Name: there_is_no_irrigation_today

Sub-type of: irrigation_based_on_eta, daily_basis_interval, suggest_ daily_interval
Properties:

Name	use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: suggest_daily_interval_pcf (page 55))
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	value
Description	The value of the weekly irrigation interval.
Source of Value	Derived from function: there_is_no_irrigation_today_f (page 71)
Type	Real
Legal values	Number-range(0, 1000)

2.10 Computational Ontology: *Water_Requirement Ontology*

2.10.1 water_requirement

Name: water_requirement
Properties:

Name	value
Description	The value of a crop's water requirement
Source of Value	Derived function: water_requirement_f (page 71)
Type	Real
Legal values	Number-range(0, 1000)

2.10.2 adaptive_lr

Name: adaptive_lr
Sub-type of: water_requirement
Properties:

Name	value
Description	The value of a crop's leaching requirements (adapted)
Source of Value	Derived from post condition function: adaptive_lr_pocf (page 55)
Type	Real
Legal values	Number-range(0, 1000)

2.10.3 lr

Name: lr
Sub-type of: water_requirement
Properties:

Name	value
Description	The value of a crop's leaching requirements
Source of Value	Derived from function: lr_f (page 71)
Type	Real
Legal values	Number-range(0, 1000)

2.10.4 ece

Name: ece
Sub-type of: lr

Properties:

Name	value
Description	The value of a crop's salt tolerance
Source of Value	Derived from post condition function ece_t (page 62)
Type	Real
Legal values	Number-range(0, 1000)

2.10.5 water_used

Name: water_used
Sub-type of: water_requirement
Properties:

Name	Value
Description	The amount of water used
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: water_used_pcf (page 55)
Type	Real
Legal values	Number-range(0, 1000)

2.10.6 water_used_weekly

Name: water_used_weekly
Sub-type of: water_used
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: water_used_pcf (page 55)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	Value
Description	The amount of water used weekly
Source of Value	Derived from function: water_used_weekly_f (page 72)
Type	Real
Legal values	Number-range(0, 1000)

2.10.7 water_used_daily

Name: water_used_daily
Sub-type of: water_used
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: water_used_pcf (page 55)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	Value
Description	The amount of water used
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: water_used_daily_pcf (page 56)
Type	Real

Legal values	Number-range(0, 1000)

2.10.8 wu_depend_on_pawc

Name: wu_depend_on_pawc
Sub-type of: water_used_daily
Properties:

Name	**Use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: water_used_daily_pcf (page 56)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**Value**
Description	???
Source of Value	Derived from function: wu_depend_on_pawc_f (page 72)
Type	Real
Legal values	Number-range(0, 1000)

2.10.9 wu_depend_on_eta_&eta_acc

Name: wu_depend_on_eta_&eta_acc
Sub-type of: water_used_daily
Properties:

Name	**Use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: water_used_daily_pcf (page 56)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**Value**
Description	???
Source of Value	Derived from function: wu_depend_on_eta&eta_acc_f (page 72)
Type	Real
Legal values	Number-range(0, 1000)

2.11 Computational Ontology: *Accumulated eta Ontology*

2.11.1 accumulated_eta

Name: accumulated_eta
Properties:

Name	**Value**
Description	The accumulated eta
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: accumulated_eta_pcf (page 56)
Type	Real
Legal values	Number-range(0, 1000)

2.11.2 State_one

Name: State_one
Sub-type of: accumulated_eta
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: accumulated_eta_pcf (page 56)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	Value
Description	???
Source of Value	Derived from function: state_one_f (page 72)
Type	Real
Legal values	Number-range(0, 1000)

2.11.3 State_two

Name: State_two
Sub-type of: accumulated_eta
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: accumulated_eta_pcf (page 56)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	Value
Description	???
Source of Value	Derived from function: state_two_f (page 72)
Type	Real
Legal values	Number-range(0, 1000)

2.11.4 State_three

Name: State_three
Sub-type of: accumulated_eta
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: accumulated_eta_pcf (page 56)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	Value
Description	
Source of Value	Derived from function: state_three_f (page 73)
Type	Real
Legal values	Number-range(0, 1000)

2.11.5 State_four

Name: State_four
Sub-type of: accumulated_eta
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node

Source of Value	Derived from relation: accumulated_eta_pcf (page 56)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**Value**
Description	
Source of Value	Derived from function: state_three_f (page 73)
Type	Real
Legal values	Number-range(0, 1000)

2.12 Computational Ontology: *Irrigation Units Ontology*

2.12.1 irrigation_units

Name: irrigation_units
Properties:

Name	**Value**
Description	Te value of irrigation units?
Source of Value	Derived from a dependency graph. The node used to derive this value is the pre-condition relation: irrigation_units_pcf (page 57)
Type	Real
Legal values	Number-range(0, 1000)

Name	**Value_n**
Description	The type of irrigation to be applied
Source of Value	Derived from relation: irrigation_type_r (page 57)
Type	Nominal
Legal values	light_irrigation, medium_irrigation, heavy_irrigation

Documenter's comment, why is this(value_n put here?

2.12.2 flooding_irrigation

Name: flooding_irrigation
Sub-type of: irrigation_units
Properties:

Name	**Use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: irrigation_units_pcf (page 57)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**Value**
Description	
Source of Value	Aggregation
Type	Real
Legal values	Number-range(0, 1000)

2.12.3 irrigation_type

Name: irrigation_type
Sub-type of: flooding_irrigation
Properties:

Name	Value_n
Description	
Source of Value	Derived from relation: irrigation_type_r (page 57)
Type	Real
Legal values	Number-range(0, 1000)

2.12.4 drip_irrigation

Name: drip _irrigation
Sub-type of: irrigation_units
Properties:

Name	Use
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: irrigation_units_pcf (page 57)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	Value
Description	
Source of Value	Aggregation
Type	Real
Legal values	Number-range(0, 1000)

2.12.5 motors_hours_work

Name: motors_hours_work
Sub-type of: drip _irrigation
Properties:

Name	Value
Description	The number of hours to operate the motor
Source of Value	Function: motors_hours_work_f (page 73)
Type	Real
Legal values	Number-range(0, 1000)

2.12.6 no_of_irrigation_during_day

Name: no_of_irrigation_during_day
Sub-type of: drip _irrigation
Properties:

Name	Value
Description	The number of irrigation operations to be performed during the day
Source of Value	Relation: no_of_irrigation_during_day_r (page 57)
Type	Real
Legal values	Number-range(0, 1000)

2.12.7 intake

Name: intake
Sub-type of: no_of_irrigation_during_day
Properties:

Name	Value
Description	The value of the soil intake rate
Source of Value	Derived from a dependency graph. The node used to derive this

	value is the pre-condition relation: intake_pcf (page 57)
Type	Real
Legal values	Number-range(0, 1000)

2.12.8 intake_l

Name: intake_l
Sub-type of: intake
Properties:

Name	**Use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: intake_pcf (page 57)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**Value**
Description	The value of the soil intake when soil.sp is low
Source of Value	Function: intake_l_f (page 73)
Type	Real
Legal values	Number-range(0, 1000)

2.12.9 intake_h

Name: intake_h
Sub-type of: intake
Properties:

Name	**Use**
Description	A flag to indicate whether or not to use this node
Source of Value	Derived from relation: intake_pcf (page 57)
Type	Nominal /Single
Legal values	'yes', 'no' no' (default = 'no')

Name	**Value**
Description	The value of the soil intake when soil.sp is high
Source of Value	Function: intake_h_f (page 73)
Type	Real
Legal values	Number-range(0, 1000)

3 Required Relationships

In the following sub-sections, the list of relationships that are required for using the generic irrigation model, will be detailed. For each relation, a description as well as the required output will be described. Though, in some cases an input might also be outlined, the expert system designer is free not to use that particular input, or to use other inputs as long as the specified output results from the relation.

3.1 Plantation Related (*Plantation Ontology)*

3.1.1 Soil_type

Relation name: Soil_type
Type: Built-in
Description:

> *This relation is used to determine the value of soil type which can be fine, coarse or medium. This relation is a built in relation. It is strongly adviced that it be left as is:*

Relation input: soil.texture
Relation output: soil.type
Modifiable as long as it returns an appropriate soil.type
Relation Body:

	Condition	Action
R1	If soil.texture == (‘clay’ or ‘clay loam’ or ‘silty clay’ or ‘silty clay loam’)	Then soil.type = fine
R2	If soil.texture == (‘sandy clay’ or ‘sandy clay loam’ or ‘silt loam’ or ‘silty loam’)	Then soil.type = medium
R3	If soil.texture == (‘sandy loam’ or ‘sand’ or ‘loamy sand’)	Then soil.type = coarse

3.1.2 growth_stages

Relation name: growth_stages
*Type: **User-defined***
Description:

> *This relation is used to determine the number of days in each of the plant’s growth stages.*

Relation input: *whatever is appropriate*
Relation output: plant.init_stage, plant.ve_stage, plant.fl_stage, plant.fr_stage

3.1.3 get_crop_specific_basic_data

Relation name: get_crop_specific_basic_data
*Type: **User-defined***
Description:

> *This relation is used to initialize any basic data that is required for the initial calculations of the irrigation operation for a specific crop*

Relation input: *whatever is appropriate*
Relation output: *whatever is appropriate*

3.2 Operations Related (*Operations Ontology)*

3.2.1 tunnel_ painting _factor_r

Relation name: tunnel_ painting _factor_r
Type: Built-in
Description:

> *This relation is used to determine the tunnel painting factor*

Relation input: task_parameters.current_month, tunnel_ painting. done
Relation output: tunnel_ painting. tunnel_ painting _factor

Modifiable as long as it returns an appropriate tunnel painting factor

Relation Body:

	Condition	Action
R1	If tunnel_painting.done == 'yes' & task_parameters. current_month == (2 or 3 or 4 or 5 or 6 or 7 or 8 or 9)	Then tunnel_ painting. tunnel_ painting _factor = 0.9
R2	If tunnel_painting.done == 'yes' & task_parameters. current_month != (2 or 3 or 4 or 5 or 6 or 7 or 8 or 9)	Then tunnel_ painting. tunnel_ painting _factor = 1

3.3 Computational Ontology: Et0 Ontology

3.3.1 et0_pcf

Relation name: et0_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which method to use in calculating the value of et0.*

Relation input: farm. planting_type
Relation output: et0_hargerve.use, et0_pennman.use
Not Modifiable…

Relation Body:

	Condition	Action
R1	if farm. planting_type == High_tunnel	et0_hargerve.use = yes
R2	if farm. planting_type == (Open_field or Low_tunnel)	et0_ penman = yes

3.3.2 et0_hargerve_pcf

Relation name: et0_hargerve _pcf
Type: Built-in
Description:

> *When the hargerve method is used, this relation serves as a pre-condition function to determine method of calculation to follow based on the required type of schedule*

Relation input: irrigation.schedule_type
Relation output: et0_har_c.use, smooth_et0_hargerve.use
Not Modifiable…

Relation Body:

	Condition	Action
R1	if irrigation.schedule_type == daily	et0_har_c.use = yes
R2	if irrigation.schedule_type == weekly	smooth_et0_ hargerve.use = yes

3.3.3 et0_pennman_pcf

Relation name: et0_pennman _pcf
Type: Built-in
Description:

> *When the pennman method is used, this relation serves as a pre-condition function to determine method of calculation to follow based on???*

Relation input: task_parameters.next_month, climate.max_rh
Relation output: l_rh_c.use, m_rh_c.use, h_rh_c.use
Modifiable…
Relation Body:

	Condition	Action
R1	if climate.max_rh(task_parameters.next_month) <= 30	l_rh_c .use = yes
R2	if climate.max_rh(task_parameters.next_month) > 30 && climate.max_rh(task_parameters.next_month) <=60	m_rh_c.use = yes
R3	if climate.max_rh(task_parameters.next_month) > 60	h_rh_c.use = yes

3.3.4 control_f_r

Relation name: control_f_r
Type: Built-in
Description:

> *This relation is used to derive the value of the control factor control_f*

Relation input: task_parameters. current_day
Relation output: control_f.value
Modifiable as values can be crop dependant
Relation Body:

	Condition	Action
R1	if task_parameters. current_day < 8	control_f.value= 0
R2	if task_parameters. current_day >7 && task_parameters. current_day < 15	control_f.value = 1
R3	if task_parameters. current_day >14 && task_parameters. current_day < 22	control_f.value = 2
R4	if task_parameters. current_day >21	control_f.value = 3

3.4 Computational Ontology: Eta Ontology

3.4.1 gc_pcf

Relation name: gc_pcf
Type: Built-in
Description:

> *The green covering area is calculated differently for each of a crop/plant's growth stages. This relation serves as a pre-condition function to determine which node to follow in order to calculate this value, where a node is designated for each of the growth stages.*

Relation input: task_parameters.age, plant. init_stage, plant. init_ve_stage, plant. init_ve_fl_stage , current_planting. agriculture_method
Relation output: gc_init_st.use, gc_ve_st.use, gc_fl_st.use, gc_fr_st.use
Not Modifiable..
Relation Body:

	Condition	Action
R1	if task_parameters. age <= (plant. init_stage) && current_planting. agriculture_method = seedling	gc_init_st .use = yes
R2	if task_parameters..age > (plant. init_stage) && task_parameters..age <= (plant. init_ve_stage)	gc_ve_st.use = yes
R3	if task_parameters..age > (plant.init_ve_stage) && task_parameters..age <= (plant. init_ve_fl_stage)	gc_fl_st.use = yes

R4	if task_parameters..age > (plant. init_ve_fl_stage)	gc_fr_st.use = yes

Documenter's note:What if the current_planting. agriculture_method is not equal to seeding?

3.4.2 kc_pcf

Relation name: kc_pcf
Type: Built-in
Description:

> *A crop coefficient kc is calculated differently for each of a crop/plant's growth stages. This relation serves as a pre-condition function to determine which node to follow in order to calculate this value, where a node is designated for each of the growth stages.*

Relation input: task_parameters.age, plant. init_stage, plant. init_ve_stage, plant. init_ve_fl_stage , current_planting. agriculture_method
Relation output: kc_init_st .use, kc_ve_st.use, kc_fl_st.use, kc_fr_st.use
Not Modifiable..
Relation Body:

	Condition	Action
R1	if task_parameters. age <= (plant. init_stage) && current_planting. agriculture_method = seedling	kc_init_st .use = yes
R2	if task_parameters..age > (plant. init_stage) && task_parameters..age <= (plant. init_ve_stage)	kc_ve_st.use = yes
R3	if task_parameters..age > (plant.init_ve_stage) && task_parameters..age <= (plant. init_ve_fl_stage)	kc_fl_st.use = yes
R4	if task_parameters..age > (plant. init_ve_fl_stage)	kc_fr_st.use = yes

3.4.3 kc_fl_st _pcf

Relation name: kc_fl_st _pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the crop coefficient kc when the plant is in the flowering stage. This depends on the type of environment in which planning takes place as denoted by the farm type.*

Relation input: farm. planting_type
Relation output: kc_fl_h_and_lt.use, kc_fl_of.use

Not Modifiable

Relation Body:

	Condition	Action
R1	if farm. planting_type = 'High_tunnel' or farm. planting_type = 'Low_tunnel'	kc_fl_h_and_lt.use = yes
R2	if farm. planting_type = 'Open_field'	kc_fl_of.use = yes

3.4.4 kc_fr_st _pcf

Relation name: kc_fr_st _pcf
Type: Built-in
Description:

This relation serves as a pre-condition function to determine which node to follow in order to calculate the crop coefficient kc when the plant is in the fruition stage. This depends on the type of environment in which planning takes place as denoted by the farm type.

Relation input: farm. planting_type
Relation output: kc_fr_h_and_lt.use, kc_fr_of.use

Not Modifiable

Relation Body:

	Condition	Action
R1	if farm. planting_type = 'High_tunnel' or farm. planting_type = 'Low_tunnel'	kc_fr_h_and_lt.use = yes
R2	if farm. planting_type = 'Open_field'	kc_fr_of.use = yes

3.4.5 kc_fr_of _pcf

Relation name: kc_fr_of _pcf
Type: Built-in
Description:

This relation serves as a pre-condition function to determine which node to follow in order to calculate the crop coefficient kc when the plant is in the fruition stage and is being planted in an open field. This value depends on the age of the plant.

Relation input: task_parameters.age
Relation output: kc_fr_l_age.use, kc_fr_g_age.use

Modifiable: The age threshold used may vary from one plant to another.

Relation Body:

	Condition	Action
R1	if task_parameters. age <= 160	kc_fr_l_age.use.use = yes
R2	if task_parameters. age >160	kc_fr_g_age.use = yes

3.4.6 depression_factor_r

Relation name: depression_factor_r
Type: Built-in
Description:

This relation is used to calculate the value of the depression factor.

Relation input: farm. planting_type (or whatever is appropriate)
Relation output: depression_factor.value

Modifiable: Whatever factors that affect the depression factor should be included in the rules that calculate the factor. These may vary from plant to plant.

3.4.7 unit_factor_r

Relation name: unit_factor_r
Type: Built-in
Description:

This relation is used to calculate the value of the unit factor.

Relation input: farm. planting_type, farm.area (and/or whatever is appropriate)
Relation output: unit_factor.value

Modifiable: Whatever factors that affect the unit factor should be included in the rules that calculate the factor. These may vary from plant to plant.

Relation Body:

	Condition	Action
R1	if farm. planting_type = 'Open_field' or farm. planting_type = 'Low_tunnel'	unit_factor.value = 4200
R2	if farm. planting_type = 'High_tunnel'	unit_factor.value = farm.area

3.5 Computational Ontology: Pawc Ontology

3.5.1 rd_pcf

Relation name: rd_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the value of a crop's root depth, depending on the growth stage*

Relation input: task_parameters.age, plant. init_stage, plant. init_ve_stage, plant. init_ve_fl_stage , current_planting. agriculture_method
Relation output: rd_init_st .use, rd_ve_st.use, rd_fl_st.use, rd_fr_st.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if task_parameters. age <= (plant. init_stage) && current_planting. agriculture_method = seedling	rd_init_st .use = yes
R2	if task_parameters..age > (plant. init_stage) && task_parameters..age <= (plant. init_ve_stage)	rd_ve_st.use = yes
R3	if task_parameters..age > (plant.init_ve_stage) && task_parameters..age <= (plant. init_ve_fl_stage)	rd_fl_st.use = yes
R4	if task_parameters..age > (plant. init_ve_fl_stage)	rd_fr_st.use = yes

3.5.2 rd_init_st_pcf

Relation name: rd_init_st_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the value of a crop's root depth in the initiation stage, depending on the irrigation method*

Relation input: irrigation.method
Relation output: rd_init_st_d .use, rd_init_st_f .use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if irrigation. irrigation.method = drip	rd_init_st_d .use = yes
R2	if irrigation. irrigation.method = flooding	rd_init_st_f .use = yes

3.5.3 rd_ve_st_pcf

Relation name: rd_ve_st_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the value of a crop's root depth in the vegetative stage, depending on the irrigation method.*

Relation input: irrigation.method
Relation output: rd_ve_st_d .use, rd_ve_st_f .use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if irrigation. irrigation.method = drip	rd_ve_st_d .use = yes
R2	if irrigation. irrigation.method = flooding	rd_ve_st_f .use = yes

3.5.4 rd_fl_st_pcf

Relation name: rd_fl_st_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the value of a crop's root depth in the flowering stage, depending on the irrigation method.*

Relation input: irrigation.method
Relation output: rd_fl_st_d .use, rd_fl_st_f .use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if irrigation. irrigation.method = drip	rd_fl_st_d .use = yes
R2	if irrigation. irrigation.method = flooding	rd_fl_st_f .use = yes

3.5.5 rd_fr_st_pcf

Relation name: rd_fr_st_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the value of a crop's root depth in the fruiting stage, depending on the irrigation method.*

Relation input: irrigation.method
Relation output: rd_fr_st_d .use, rd_fr_st_f .use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if irrigation. irrigation.method = drip	rd_fr_st_d .use = yes
R2	if irrigation. irrigation.method = flooding	rd_fr_st_f .use = yes

3.5.6 sp_factor_pcf

Relation name: rd_fr_st_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the value of the soil saturated percentage .*

Relation input: soil.sp
Relation output: sp_l.use, sp_g.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if soil.sp sp < *37*	sp_l.use = yes
R2	if soil.sp sp >= *37*	sp_g.use = yes

3.6 Computational Ontology: Interval Ontology

3.6.1 interval_pcf

Relation name: interval_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the irrigation interval .*

Relation input: irrigation.schedule_type
Relation output: weekly_basis_interval.use, daily_basis_interval.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	if irrigation.schedule_type = weekly	weekly_basis_interval.use = yes
R2	if irrigation.schedule_type = daily	daily_basis_interval.use = yes

3.6.2 weekly_basis_interval_pcf

Relation name: weekly_basis_interval_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the irrigation interval .*

Relation input: irrigation.controled_water
Relation output: interval_weekly.use, user_suggested_interval_weekly.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	irrigation.controled_water= yes	interval_weekly.use = yes
R2	irrigation.controled_water= no	user_suggested_interval_weekly.use = yes

3.6.3 daily_basis_interval_pcf

Relation name: daily_basis_interval_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the irrigation interval .*

Relation input: irrigation.controled_water, task_parameters.age
Relation output: there_is_irrigation_today.use, irrigation_based_on_eta.use, suggest_daily_interval.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	task_parameters.age= 1	there_is_irrigation_today.use = yes
R2	irrigation.controled_water= yes & task_parameters.age > 1	irrigation_based_on_eta.use = yes
R3	irrigation.controled_water= no & task_parameters.age > 1	suggest_daily_interval .use = yes

3.6.4 irrigation_based_on_eta_pcf

Relation name: irrigation_based_on_eta_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the irrigation interval .*

Relation input: task_parameter. eta_plus_eta_acc , task_parameters.pawc_decrease_twenty_per
Relation output: there_is_irrigation_today.use, there_is_no_rrigation_today.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	task_parameter. eta_plus_eta_acc >= task_parameters. pawc_decrease_twenty_per	there_is_irrigation_today.use = yes
R2	task_parameter. eta_plus_eta_acc < task_parameters. pawc_decrease_twenty_per	there_is_no_irrigation_today.use = yes

3.6.5 suggest_daily_interval_pcf

Relation name: suggest_daily_interval_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the irrigation interval .*

Relation input: task_parameter. interval_number
Relation output: there_is_irrigation_today.use, there_is_no_rrigation_today.use

Not Modifiable..

Relation Body:

	Condition	Action
R1	irrigation.user_suggested_interval = task_parameters. interval_number	there_is_irrigation_today.use = yes
R2	irrigation.user_suggested_interval \== task_parameters. interval_number	there_is_no_irrigation_today.use = yes

3.7 Computational Ontology: *Water_Requirement Ontology*

3.7.1 adaptive_lr_pocf

Relation name: lr_pocf
Type: Built-in
Description:

> *This relation serves as a post-condition function to set the value of lr (leaching requirment)*

Relation input: lr.value
Relation output: adaptive_lr.value

Modifiable

Relation Body:

	Condition	Action
R1	if lr.value> 0.25	lr.value =0.25

3.7.2 water_used_pcf

Relation name: water_used_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the amount of water used.*

Relation input: irrigation.schedule_type
Relation output: water_used_daily.use, water_used_weekly.use

Not Modifiable

Relation Body:

	Condition	Action
R1	if irrigation.schedule_type = daily	water_used_daily.use = yes

R2	if irrigation.schedule_type = weekly	water_used_weekly.use = yes

3.7.3 water_used_daily_pcf

Relation name: water_used_daily_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the amount of water used when age is used*

Relation input: task_parameters.age, irrigation.controled_water
Relation output: wu_depend_on_pawc.use, wu_depend_on_eta&eta_acc.use

Not Modifiable

Relation Body:

	Condition	**Action**
R1	if task_parameters.age = 1	wu_depend_on_pawc.use = yes
R2	if task_parameters.age > 1 && irrigation.controled_water = yes	wu_depend_on_pawc.use = yes
R3	if task_parameters.age > 1 && if irrigation.controled_water = no	wu_depend_on_eta&eta_acc.use= yes

3.8 Computational Ontology: *Accumulated eta Ontology*

3.8.1 accumulated_eta _pcf

Relation name: accumulated_eta _pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the accumulated eta*

Relation input: interval.value task_parameters.age, irrigation.controled_water
Relation output: state_one.use, state_two.use, state_three.use, state_four.use

Not Modifiable

Relation Body:

	Condition	**Action**
R1	if interval.value = 0	state_one.use = yes
R2	if interval.value =1 && task_parameters.age = 1	state_two.use = yes
R3	interval.value =1 && task_parameters.age > 1 && irrigation.controled_water = yes	state_three.use = yes
R4	interval.value =1 && task_parameters.age > 1 && irrigation.controled_water = no	state_four.use = yes

3.9 Computational Ontology: *Irrigation Units Ontology*

3.9.1 Irrigation_units_pcf

Relation name: Irrigation_units_pcf
Type: Built-in
Description:

> *This relation serves as a pre-condition function to determine which node to follow in order to calculate the irrigation units??*

Relation input: irrigation.method
Relation output: flooding_irrigation.use, drip_irrigation.use

Not Modifiable

Relation Body:

	Condition	Action
R1	if irrigation.method= flooding	flooding_irrigation.use = yes
R2	if irrigation.method= drip	drip_irrigation.use = yes

3.9.2 Irrigation_type_r

Relation name: Irrigation_type_r
Type: Built-in
Description:

> *This relation is used to determine the type of irrigation to be applied based on the calculated water requirements*

Relation input: water_requirement.value
Relation output: irrigation_type.value_n

Modifiable

Relation Body:

	Condition	Action
R1	if water_requirement.value >= 150 && water_requirement.value <= 200	irrigation_type.value_n= light_irrigation
R2	if water_requirement.value > 200 && water_requirement.value <= 300	irrigation_type.value_n= medium_irrigation
R3	if water_requirement.value > 300	irrigation_type.value_n= heavy_irrigation

3.9.3 no_of_irrigation_during_day_r

Relation name: no_of_irrigation_during_day_r
Type: Built-in
Description:

> *This relation is used to determine the number of irrigation operations to be performed during the day*

Relation input: water_requirement.value, intake.value
Relation output: no_of_irrigation_during_day.value

Modifiable

Relation Body:

	Condition	Action
R1	if intake.value < water_requirement.value	no_of_irrigation_during_day.value= 2
R2	if intake.value >= water_requirement.value	no_of_irrigation_during_day.value= 1

3.9.4 intake_pcf

Relation name: intake_pcf
Type: Built-in
Description:

> *This relation is used to determine the number of irrigation operations to be performed during the day*

Relation input: soil.sp
Relation output: intake_l.use, intake_h.use

Modifiable

Relation Body:

	Condition	Action
R1	if soil.sp <= 41.7	intake_l.use = yes
R2	if soil.sp > 41.7	intake_h.use = yes

4 Required Tables

In the following sub-sections, the list of relationships that are required for using the generic irrigation model, will be detailed. For each relation, a description as well as the required output will be described. Though, in some cases an input might also be outlined, the expert system designer is free not to use that particular input, or to use other inputs as long as the specified output results from the relation.

4.1 Plantation Related (*Plantation Ontology)*

4.1.1 SP_table

Table name: SP_table
Description:

> *This table is used to calculate the soil saturation percentage*

Table input: soil.type
Table output: soil.sp
Table:

Input	Output
soil.type	**soil.sp**
Fine	25
medium	22
coarse	20

4.1.2 SBD_table

Table name: SBD_table
Description:

> *This table is used to calculate the soil bulk density*

Table input: soil.type
Table output: soil.sbd
Table:

Input	Output
soil.type	**soil.sbd**
Fine	1.3
medium	1.35
coarse	1.4

4.1.3 msh_a_table

Table name: msh_a_table
Description:

> *This table is used to calculate one of the maximum sunshine hours parameters*

Table input: task_parameters.current_month
Table output: climate.msh_par_a
Table:

Input	Output
Task_parameters.current_month	**climate.msh_par_a**
1	12.0989
2	12.0989
3	12.01444
4	12.00222
5	11.97556
6	11.9467
7	11.9467
8	12.00667
9	12.01556
10	12.06222
11	12.1322
12	12.1322

4.1.4 msh_b_table

Table name: msh_b_table
Description:

> *This table is used to calculate one of the maximum sunshine hours parameters*

Table input: task_parameters.current_month
Table output: climate.msh_par_a
Table:

Input	Output
Task_parameters.current_month	climate.msh_par_b
1	- 0.0455
2	- 0.0455
3	- 0.00233
4	0.031
5	0.057333
6	0.06933
7	0.06933
8	0.041333
9	- 0.012
10	- 0.02033
11	- 0.0588
12	- 0.0588

4.1.5 ra_a_table

Table name: ra_a_table
Description:

> *This table is used to calculate one of the solar radiation parameters*

Table input: task_parameters.current_month
Table output: climate.ra_par_a
Table:

Input	Output
Task_parameters.current_month	climate.ra_par_a
1	15.26147
2	15.89913
3	16.18918
4	15.76883
5	14.708
6	14.708
7	14.708
8	14.708
9	15.84288
10	15.95455
11	15.49091
12	15.12208

4.1.6 ra_b_table

Table name: ra_b_table
Description:

> *This table is used to calculate one of the solar radiation parameters*

Table input: task_parameters.current_month
Table output: climate.ra_par_b
Table:

Input	Output
task_parameters.current_month	climate.ra_par_b
1	- 0.21331
2	- 0.17519
3	- 0.10636
4	- 0.02416
5	0.0529
6	0.0529
7	0.0529
8	0.0529
9	- 0.06929
10	0.14701
11	0.20383
12	- 0.0588

4.2 Operations Related (*Operations Ontology)*

4.2.1 irrigation_efficiency_t

Table name: irrigation_efficiency_t
Description:

> *This table is used to calculate irrigation efficiency based on the irrigation method employed*

Table input: irrigation.method
Table output: irrigation.irrigation_efficiency
Table:

Input	Output
irrigation.method	irrigation_efficiency
drip	0.9
flooding	0.65

4.2.2 Saa_t

Table name: Saa_table
Description:

> *This table is used to calculate the soil absorbed area*

Table input: irrigation.method, soil.type
Table output: irrigation.saa
Table:

Input		Output
irrigation.method	Soil.type	irrigation.saa
flooding	fine	60
flooding	medium	70
flooding	coarse	80
drip	fine	25
drip	medium	25
drip	coarse	25

4.3 Computational Ontology: Pawc Ontology

4.3.1 ad_t

Table name: ad_t
Description:

> *This table is used to calculate the value of crop allowable water depletion*

Table input: farm.planting_type and/or inputs that apply
Table output: ad.value

4.3.2 max_rd_t

Table name: max_rd_t
Description:

> *This table is used to calculate the maximum rooting depth*

Table input: farm.planting_type, irrigation.method, soil.type and/or inputs that apply
Table output: ad.value

4.4 Computational Ontology: *Water_Requirement Ontology*

4.4.1 ece_t

Table name: ece_t
Description:

> *This table is used to calculate ece which is the salt tolerance of a plant measured in (ds/m)*

Table input: whatever is appropriate
Table output: ece.value

5 Required Functions

Some text!

5.1 Plantation Related (*Plantation Ontology)*

5.1.1 msh_f

function name: msh_f
Description:

> *This function is used to calculate the value of maximum sunshine hours*

Function input: climate.msh_par_a, climate.msh_par_b , farm.latitude
Function output: climate.msh
Function body:

Climate.msh = climate.msh_par_a + climate.msh_par_b* farm.latitude

5.1.2 ra_f

function name: ra_f
Description:

This function is used to calculate the value of extra radiation

Function input: climate.ra_par_a, climate.ra_par_b , farm.latitude
Function output: climate.ra
Function body:
climate.ra = climate.ra_par_a + climate.ra_par_b* farm.latitude

5.1.3 growth_period_f

function name: growth_period_f
Description:
This function is used to calculate the value of a plant's growth period

Function input: plant.init_stage, plant.ve_stage, plant.fl_stage, plant.fr_stage
Function output: plant.growth_period
Function body:
plant.growth_period = plant.init_stage+ plant.ve_stage+ plant.fl_stage+ plant.fr_stage

5.1.4 init_ve_stage_f

function name: init_ve_stage_f
Description:
This function is used to calculate the value of a plant'si Initiation and vegetative stages

Function input: plant.init_stage, plant.ve_stage
Function output: plant.init_ve_stage
Function body:
plant.init_ve_stage= plant.init_stage+ plant.ve_stage

5.1.5 init_ve_fl_stage_f

function name: init_ve_fl_stage_f
Description:
This function is used to calculate the value of a plant's initiation, vegetative and flowering stages

Function input: plant.init_stage, plant.ve_stage, plant.fl_stage
Function output: plant.init_ve_fl_stage
Function body:
plant.init_ve_fl_stage = plant.init_stage+ plant.ve_stage+ plant.fl_stage

5.2 Computational Ontology: Et0 Ontology

5.2.1 et0_ smooth _hargerve_f

function name: et0_ smooth _hargerve_f
Description:
This function is used to calculate the value of et0_ smooth _hargerve

Function input: et0_har_c.value, et0_har_n.value, control_f.value
Function output: et0_ smooth _hargerve.value
Function body:
smooth_et0_hargerve.value= et0_har_c.value+ et0_har_n.value *control_f.value

5.2.2 et0_har_c_f

function name: et0_har_c_f
Description:

> *This function is used to calculate the value of et0_har_c*

Function input: climate related properties: avg_rh, avg_tc, ash , msh,, ra
Function output: et0_har_c.value
Function body:

et0_har_c.value=0.0075*(0.097-0.00042* climate. avg_rh)*(climate.avg_tc *1.8+32)*sqrt(100* climate.ash /climate. msh)* climate. ra

5.2.3 et0_har_n_f

function name: et0_har_n_f
Description:

> *This function is used to calculate the value of et0_har_n*

Function input: climate related properties: avg_rh_next_month, avg_tc_next_month, ash_next_month, msh_next_month, ra_next_month
Function output: et0_har_n.value
Function body:

et0_har_n.value= 0.0075*(0.097-0.00042* climate. avg_rh_next_month)*(climate. avg_tc_next_month *1.8+32)*sqrt(100* climate. ash_next_month /climate. msh_next_month)* climate. ra_next_month

5.2.4 l_rh_c_f

function name: l_rh_c_f
Description:

> *This function is used to calculate the value of l_rh_c_f*

Function input: climate related properties: (ra, ash , msh, dws, nws,mws, avg_rh, avg_tc), and farm.altitude
Function output: l_rh_c.value
Function body:

l_rh_c.value= ((climate. ra *(0.25+0.5*(climate.ash / climate. msh)))*(((climate. dws/3.6)*(0.003872-0.000364*(climate. dws / climate. nws)))+(0.000133*(climate. dws / climate. nws)+0.007167))+((climate. dws /3.6)*(0.011283*(climate. dws / climate. nws)-0.08775))+(1.0425-0.0185*(climate. dws / climate. nws))*((0.440904+0.0000319* farm. altitude + climate.avg_tc *(0.011221-0.00000048* farm. altitude))*(0.75* climate. ra *(0.25+0.5*(* climate. ash /(* climate. msh)))-((10.8158+* climate. avg_tc *0.1965)*(0.34-0.044*exp((1*exp(1.872543+0.06237 climate. avg_)** climate. avg_rh /100),0.5))*(0.1+0.9*(* climate. ash /(* climate. msh)))))+(1-(0.440904+0.0000319* farm. altitude + climate. avg_tc of climate*(0.011221-0.00000048* farm. altitude)))*(0.27+0.0648* climate. mws)*(exp(1.872543+0.06237* climate. avg_tc)*(1- climate. avg_rh/100))))**

5.2.5 m_rh_c_f

function name: m_rh_c_f
Description:

> *This function is used to calculate the value of m_rh_c_f*

Function input: climate related properties: (ra, ash , msh, dws, nws,mws, avg_rh, avg_tc), and farm.altitude
Function output: m_rh_c.value
Function body:

m_rh_c.value= (climate. ra *(((climate. dws /3.6)*(0.003428-0.000022*(climate. dws / climate. nws)))+(0.00117*(climate. dws / climate. nws)+0.001167))+((climate. dws /3.6)*(0.010033*(climate. dws / climate. nws)-0.072583))+(1.125-0.0255*(climate. dws / climate. nws))*((0.440904+0.0000319* farm. altitude + climate. avg_tc *(0.011221-0.00000048* farm. altitude))*(0.75* climate. ra *(0.25+0.5*(climate. ash /(climate. msh)))-((10.8158+ climate. avg_tc *0.1965)*(0.34-0.044*exp((1*exp(1.872543+0.06237* climate. avg_tc)* climate. avg_rh /100),0.5))*(0.1+0.9*(climate. ash /(climate. msh)))))+(1-(0.440904+0.0000319*farm. altitude + climate.avg_tc *(0.011221-0.00000048* farm. altitude)))*(0.27+0.0648* climate.mws)*(exp(1.872543+0.06237* climate.avg_tc)*(1- climate.avg_rh /100))))

5.2.6 h_rh_c_f

function name: h_rh_c_f
Description:

> *This function is used to calculate the value of h_rh_c_f*

Function input: climate related properties: (ra, ash , msh, dws, nws,mws, avg_rh, avg_tc), and farm.altitude
Function output: h_rh_c.value
Function body:

h_rh_c.value= ((climate.ra *(0.25+0.5*(climate.ash / climate.msh)))*(((climate.dws /3.6)*(0.000139*(climate.dws / climate.nws)+0.003122))+(0.00185*(climate.dws / climate.nws)-0.000083))+((climate.dws /3.6)*(0.009333*(climate.dws / climate.nws)-0.0645))+(1.16625-0.026*(climate.dws / climate.nws)))*((0.440904+0.0000319*farm.altitude+ climate.avg_tc *(0.011221-0.00000048*farm.altitude))*(0.75* climate.ra *(0.25+0.5*(climate.ash /(climate.msh)))-((10.8158+ climate.avg_tc *0.1965)*(0.34-0.044*exp((1*exp(1.872543+0.06237* climate.avg_tc)* climate.avg_rh /100),0.5))*(0.1+0.9*(climate.ash /(climate.msh)))))+(1-(0.440904+0.0000319*farm.altitude+ climate.avg_tc *(0.011221-0.00000048*farm.altitude)))*(0.27+0.0648* climate.mws)*(exp(1.872543+0.06237* climate.avg_tc)*(1- climate.avg_rh /100)))

5.3 Computational Ontology: Eta Ontology

5.3.1 eta_f

function name: eta_f
Description:

> *This function is used to calculate the value of eta*

Function input: depression_factor.value, unit_factor.value, kc.value, et0.value, gc.value
Function output: eta.value
Function body:

eta.value= depression_factor.value*(unit_factor.value/1000) kc.value*et0.value*(gc.value/100)

5.3.2 gc_init_st_f

function name: gc_init_st_f
Description:

> *This function is used to calculate the value of gc_init_st*

Function input: task_parameters.age, plant. init_stage
Function output gc_init_st.value
Function body:

gc_init_st.value = 100*0.05+ task_parameters.age*100*0.5/ plant. init_stage

5.3.3 gc_ve _st_f

function name: gc_ve _st_f
Description:

> *This function is used to calculate the value of gc_ve_st*

Function input: task_parameters.age, plant. init_stage, plant. ve_stage
Function output gc_ve_st.value
Function body:

gc_ve_st.value = 100*0.55+(task_parameters.age-plant. init_stage)*100*0.4/ plant. ve_stage

5.3.4 gc_fl_st_f

function name: gc_fl _st_f
Description:

> *This function is used to calculate the value of gc_fl_st*

Function input: task_parameters.age, plant. init_stage, plant. ve_stage, plant. fl_stage
Function output: gc_fl_st.value
Function body:

gc_fl_st.value = 100*0.95+(task_parameters.age- plant. init_stage - plant. ve_stage)*100*0.05/ plant. fl_stage

5.3.5 gc_fr_st_f

function name: gc_fr _st_f
Description:

> *This function is used to calculate the value of gc_rl_st*

Function input:
Function output gc_fr_st.value
Function body:

gc_fr_st.value = 100

5.3.6 kc_init_st_f

function name: kc_init_st_f
Description:

> *This function is used to calculate the value of kc_init_st*

Function input: task_parameters.age
Function output: kc_init_st.value
Function body:

kc_init_st.value = 0.441267 + 0.008974 * task_parameters.age

5.3.7 kc_ve_st_f

function name: kc_ve_st_f

Description:

> *This function is used to calculate the value of kc_ve_st*

Function input: task_parameters.age
Function output: kc_ve_st.value

Function body:

kc_ve_st.value = 0.557 + 0.005 * task_parameters.age

5.3.8 kc_ fl_h_and_lt_f

function name: kc_ fl_h_and_lt_f
Description:

> *This function is used to calculate the value of kc_ fl_h_and_lt*

Function input: task_parameters.age
Function output: kc_ fl_h_and_lt.value

Function body:

kc_ fl_h_and_lt.value = 0.767 + 0.003 * task_parameters.age

5.3.9 kc_ fl_of_f

function name: kc_ fl_of_f
Description:

> *This function is used to calculate the value of kc_ fl_of*

Function input: task_parameters.age
Function output kc_ fl_of.value

Function body:

kc_ fl_of.value = 0.767 + 0.002 * task_parameters.age

5.3.10 kc_ fr_h_and_lt_f

function name: kc_ fr_h_and_lt_f
Description:

> *This function is used to calculate the value of kc_ fr_h_and_lt*

Function input: task_parameters.age
Function output kc_ fr_h_and_lt.value

Function body:

kc_ fr_h_and_lt.value = 1.41 - 0.0025 * task_parameters.age

5.3.11 kc_ fr_l_160_f

function name: kc_ fr_l_160_f
Description:

> *This function is used to calculate the value of kc_ fr_l_age*

Function input: task_parameters.age
Function output: kc_ fr_l_age.value

Function body:

kc_ fr_l_age.value = 2.147 - 0.01 * task_parameters.age

5.3.12 kc_ fr_g_160_f

function name: kc_ fr_g_160_f
Description:

> *This function is used to calculate the value of kc_ fr_g_age.*

Function input: task_parameters.age

Function output: kc_ fr_g_age.value

Function body:

kc_ fr_g_age= 2.147 – 0.01 * 161

5.3.13 pawc_f

function name: pawc_f

Description:

> *This function is used to calculate the value of pawc, which is the water available in the soil*

Function input: ad.value, rd.value, soil.sbd, sp_factor.value, unit_factor.value

Function output: pawc.value

Function body:

Pawc.value= ad.value* (rd.value/100)*(soil.sbd*(sp_factor.value/100))* unit_factor.value

5.3.14 rd_init_st_d_f

function name: rd_init_st_d_f

Description:

> *This function is used to calculate the value of rd_init_st_d, which is the root depth for flowering stage (drip irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage

Function output: rd_init_st_d.value

Function body:

rd_init_st_d.value= max_rd.value * 0.05 + task_parameters.age*max_rd.value*0.5/ plant. init_stage

5.3.15 rd_init_st_f_f

function name: rd_init_st_f_f

Description:

> *This function is used to calculate the value of rd_init_st_f, which is the root depth for flowering stage (flooding irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage

Function output: rd_init_st_f.value

Function body:

rd_init_st_f.value= max_rd.value * 0.25+ task_parameters.age*max_rd.value*2.5/ plant. init_stage

5.3.16 rd_ve_st_d_f

function name: rd_ve_st_d_f

Description:

> *This function is used to calculate the value of rd_ve_st_d which is the root depth for vegetative stage (drip irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage, plant. ve_stage

Function output: rd_ve_st_d.value

Function body:

rd_ve_st_d.value= max_rd.value * 0.55 + (task_parameters.age- plant. init_stage) *max_rd.value*0.4/ plant. ve_stage

5.3.17 rd_ve_st_f_f

function name: rd_ve_st_f_f

Description:

> *This function is used to calculate the value of rd_ve_st_f which is the root depth for vegetative stage (flooding irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage, plant. ve_stage
Function output: rd_ve_st_f.value

Function body:

rd_ve_st_f.value= max_rd.value * 2.75+ (task_parameters.age- plant. init_stage) *max_rd.value*2/ plant. ve_stage

5.3.18 rd_fl_st_d_f

function name: rd_fl_st_d_f
Description:

> *This function is used to calculate the value of rd_fl_st_d, which is the root depth for flowering stage (drip irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage, plant. fl_stage
Function output: rd_fl_st_d.value

Function body:

rd_fl_st_d.value= max_rd.value * 0.95 + (task_parameters.age- plant. init_stage- plant. ve_stage) *max_rd.value*0.05/ plant. fl_stage

5.3.19 rd_fl_st_f_f

function name: rd_fl_st_f_f
Description:

> *This function is used to calculate the value of rd_fl_st_d, which is the root depth for flowering stage (flooding irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage, plant. fl_stage
Function output: rd_fl_st_f.value

Function body:

rd_fl_st_d.value= max_rd.value * 4.75 + (task_parameters.age- plant. init_stage- plant. ve_stage) *max_rd.value*0.25/ plant. fl_stage

5.3.20 rd_fr_st_d_f

function name: rd_fr_st_d_f
Description:

> *This function is used to calculate the value of rd_fr_st_d, which is the root depth for flowering stage (drip irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage, plant. fl_stage
Function output: rd_fr_st_d.value

Function body:

DEFINTION FOR FUNCTION NOT FOUND!

5.3.21 rd_fr_st_f_f

function name: rd_fr_st_f_f
Description:

> *This function is used to calculate the value of rd_fr_st_d, which is the root depth for flowering stage (flooding irrigation)*

Function input: max_rd.value, task_parameters.age, plant. init_stage, plant. fl_stage
Function output: rd_fr_st_f.value

Function body:

DEFINTION FOR FUNCTION NOT FOUND!

5.3.22 sp_l_f

function name: sp_l_f
Description:

> *This function is used to calculate the value of sp_l , which is soil saturated percentage lower than 37%*

Function input: soil.sp
Function output: sp_l.value
Function body:

sp_l.value= 0.746917*0.8*sp of soil-7.09062

5.3.23 sp_h_f

function name: sp_h_f
Description:

> *This function is used to calculate the value of sp_h , which is soil saturated percentage greater than 37%*

Function input: soil.sp
Function output: sp_h.value
Function body:

sp_h.value= 33.9- 0.4*soil.sp

5.4 Computational Ontology: Interval Ontology

5.4.1 interval_weekly_f

function name: interval_weekly_f
Description:

> *This function is used to calculate the irrigation weekly intervals*

Function input: pawc.value, eta.value
Function output: interval_weekly.value
Function body:

interval_weekly.value= pawc.value/eta.value

5.4.2 user_suggested_interval_weekly_f

function name: user_suggested_interval_weekly_f
Description:

> *This function is used set a user suggested interval*

Function input: irrigation.user_suggested_interval
Function output: user_suggested_interval_weekly.value
Function body:

user_suggested_interval_weekly.value= irrigation.user_suggested_interval

5.4.3 there_is_irrigation_today_f

function name: there_is_irrigation_today_f
Description:

> *This function is used set the value of a variable which determines whether irrigation is to be carried out on a given day or not.*

Function input: no input
Function output: there_is_irrigation_today.value

Function body:

there_is_irrigation_today.value= 1

5.4.4 there_is_no_irrigation_today_f

function name: there_is_no_irrigation_today_f
Description:

> *This function is used set the value of a variable which determines whether irrigation is to be carried out on a given day or not.*

Function input: no input
Function output: there_is_no_irrigation_today.value
Function body:

there_is_no_irrigation_today.value= 0

5.4.5 eta_plus_eta_acc_f

function name: eta_plus_eta_acc_f
Description:

> *This function add current eta to accumulated eta.*

Function input: eta.value, task_parameters. previous_accumulated_eta
Function output: task_parameters .eta_plus_eta_acc
Function body:

task_parameters .eta_plus_eta_acc= eta.value+ task_parameters. previous_accumulated_eta

5.4.6 pawc_decrease_twenty_per_f

function name: pawc_decrease_twenty_per_f
Description:

> *Calculates the value of pawc reduced by 20%.*

Function input: pawc.value
Function output: task_parameters.pawc_decrease_twenty_per
Function body:

task_parameters.pawc_decrease_twenty_per = pawc.value*0.8

5.5 Computational Ontology: *Water_Requirement Ontology*

5.5.1 water_requirement_f

function name: water_requirement_f
Description:

> *Calculates the plant's water requirements*

Function input: water_used.value , irrigation.irrigation_efficiency, lr.value, plantation_factor.value
Function output: water_requirement.value
Function body:

water_requirement= ((water_used.value / irrigation.irrigation_efficiency) * (1+lr.value) * plantation_factor.value

5.5.2 lr_f

function name: lr_f
Description:

> *Calculates the value for the leaching requirement*

Function input: water.eciw, ece.value
Function output: lr.value
Function body:
lr.value=water.eciw/ece.value

5.5.3 water_used_weekly_f

function name: water_used_weekly_f
Description:
> *Calculates the weekly used water quantity*

Function input: interval.value, eta.value
Function output: water_used_weekly.value
Function body:
water_used_weekly.value=interval.value*eta.value

5.5.4 wu_depend_on_pawc_f

function name: wu_depend_on_pawc_f
Description:
> *Water used based on pawc (water available in soil)*

Function input: pawc.value
Function output: wu_depend_on_pawc.value
Function body:
wu_depend_on_pawc.value= pawc.value

5.5.5 wu_depend_on_eta&eta_acc_f

function name: wu_depend_on_eta&eta_acc_f
Description:
> *Water used based on eta and accumulated eta*

Function input: task_parameters.previous_accumulated_eta, eta.value
Function output: wu_depend_on_eta&eta_acc.value
Function body:
wu_depend_on_eta&eta_acc.value = task_parameters. previous_accumulated_eta + eta.value

5.6 Computational Ontology: *Accumulated eta Ontology*

5.6.1 State_one_f

function name: State_one_f
Description:
> *???*

Function input: task_parameters.previous_accumulated_eta, eta.value
Function output: state_one.value
Function body:
State_one.value = eta.value+ task_parameters. previous_accumulated_eta

5.6.2 State_two_f

function name: State_two_f
Description:
> *???*

Function input: eta.value
Function output: state_two.value
Function body:
State_two.value = eta.value

5.6.3 State_three_f

function name: State_three_f
Description:
???

Function input: task_parameters.previous_accumulated_eta, eta.value, pawc.value
Function output: state_three.value
Function body:
State_three.value = pawc.value –(task_parameters. previous_accumulated_eta - eta.value)

5.6.4 State_four_f

function name: State_four_f
Description:
???

Function input:
Function output: state_four.value
Function body:
State_four.value = 0

5.7 Computational Ontology: *Irrigation Units Ontology*

5.7.1 motors_hours_work_f

function name: motors_hours_work_f
Description:
The number of hours to operate the motor

Function input: water_requirement.value, irrigation.number_of_drippers, irrigation.rate_of_dripper_flow
Function output: motors_hours_work.value
Function body:
motors_hours_work.value = (water_requirement.value*60*1000) / (irrigation.number_of_drippers*irrigation.rate_of_dripper_flow)

5.7.2 intake_l_f

function name: intake_l_f
Description:
The value of the soil intake when soil.sp is low

Function input: unit_factor.value, irrigation.saa, soil.sp
Function output: intake_l.value
Function body:
intake_l.value= (unit_factor.value/1000) * (irrigation.saa/100)*(104.3753-2.31916*soil.sp)

5.7.3 intake_h_f

function name: intake_h_f
Description:

The value of the soil intake when soil.sp is high

Function input: irrigation.saa
Function output: intake_h.value
Function body:
intake_h.value= 32*irrigation.saa/100

6 Task Layer

Procedure Name	ExecuteIrr		
Description	This is the main procedure for the Irrigation Schedule		
Input	***Concept***	***Property***	***Source of value***
	farm	***ID***	***user***
	farm	***latitude***	***database***
	Current_planting	***date***	***database***
	Soil	***texture***	***database***
	irrigation	***Schedule type***	***user***
Output	None		
Pre-Conditions	The user must enter a farm ID and specify the required type of schedule to initialize this procedure		
Called Procedures	- **get_missing_data** - **irrigation_daily_schedule** - **farmIrrigation_Daily_Data** - **Revise** - **irrigation_weekly_schedule** - **output_Irrigation_schedule**		
Procedure	***Dependency Graph:*** "plantation_factors-value"		

Procedure Name	Get_missing_data		
Description	This procedure is used to initialize the basic data required for the initial calculations of the irrigation.		
Input	***Concept***	***Property***	***Source of value***
	farm	***planting_type***	***database***
	current_planting	***agriculture_method***	***database***
	irrigation	***method***	***database***
	Soil	***sp***	***database***
	Soil	***sbd***	***database***
	None		
Output	None		
Pre-Conditions	None		

Called Procedures	***Play Inference*** "soil_type"
	If (soil-sp = "") Then ***Play Inference*** "sp_t"
	If (soil-sbd = "") Then ***Play Inference*** "sbd_t"
	Play Inference "growth_stages"
	Play Inference "irrigation_efficiency_t"
	Play Inference "saa_t"		

	Play Inference "get_crop_specific_basic_data"
Procedure	

Procedure Name	**irrigation_weekly_schedule**		
Description	This is the main procedure to calculate a weekly irrigation Schedule		
Input	***Concept***	***Property***	***Source of value***
	task_parameters	***current_date***	***system***
Output	weekly Irrigation Schedule		
Pre-Conditions	Schedule type must be "Weekly"		
Called Procedures	- **initialization_irrigation_weekly_schedule** - **reset_climate_data** - **propose_irrigation_schedule** - **adjusted_irrigation_weekly_schedule**		

Procedure Name	**initialization_irrigation_weekly_schedule**		
Description	Initialize all facts and variables associated with weekly irrigation schedule task.		
Input	***Concept***	***Property***	***Source of value***
	Current_planting	***date***	***database***
Output	***Concept***	***Property***	***Source of value***
	task_parameters	***week_number***	***system***
	task_parameters	***age***	***system***
	task_parameters	***current_date***	***system***
	task_parameters	***current_month***	***system***
	task_parameters	***current_day***	***system***
	task_parameters	***next_month***	***system***
Pre-Conditions	None		
Called Procedures	None		

Procedure Name	**reset_climate_data**		
Description	To collect climate data from the data		
Input	*Concept*	*Property*	*Source of value*
	climate	***avg_tc***	***database***
	climate	***avg_rh***	***database***
	climate	***ash***	***database***
	climate	***msh***	***database***
	climate	***ra***	***database***
	climate	***max_rh***	***database***
	climate	***mws***	***database***

	climate	***dws***	***database***
	climate	***nws***	***database***
	farm	***planting_type***	***database***
	climate	***avg_tc_next_month***	***database***
	climate	***avg_rh_next_month***	***database***
	climate	***ash_next_month***	***database***
	climate	***msh_next_month***	***database***
	climate	***ra_next_month***	***database***
Output	None		
Pre-Conditions	None		
Called Procedures	None		

Procedure	If (FarmType = "high_tunnel") Then
	GetValue climate-avg_tc_next_month
	GetValue climate-avg_rh_next_month
	GetValue climate-ash_next_month
	GetValue climate-msh_next_month
	GetValue climate-ra_next_month
	End If
	If (climate-ra = "") Then
	Play Inference "ra_f"
	If (climate-msh = "") Then
	Play Inference "msh_f"		

Procedure Name	**propose_irrigation_schedule**		
Description	Get the needed value for the water requirements		
Input	***Concept***	***Property***	***Source of value***
	irrigation	***controled_water***	***database***
Output	***Concept***	***Property***	***Source of value***
	et0	***value***	***DG***
	eta	***value***	***DG***
	pawc	***value***	***DG***
	water_requirement	***value***	***DG***
Pre-Conditions	None		
Called Procedures	None		

Procedure	***Dependency Graph*** et0-value
	Dependency Graph eta-value
	If (controledwater = "yes") Then
	Dependency Graph pawc-value
	End If
	If (schedule_type = "daily" & controled_water = "yes") Then
	Play Inference "acc_eta_pawc" //inference not found!
	Dependency Graph interval-value
	Else
	Dependency Graph interval-value
	End If
	If (schedule_type = "daily") Then		

	Dependency Graph accumulated_eta-value End If

Procedure Name	**adjusted_irrigation_weekly_schedule**		
Description	To determine a new value of the irrigation schedule parameters. So, a new entry of the schedule could be started.		
Input	**Concept**	**Property**	**Source of value**
	plantation_factors	**value**	
Output	***Concept***	***Property***	***Source of value***
	task_parameters	**age**	**system**
	task_parameters	***week_number***	**system**
	task_parameters	***current_date***	**system**
	task_parameters	***current_day***	**system**
	task_parameters	**current_month**	**system**
	task_parameters	***next_month***	**system**
Pre-Conditions	The schedule type is weekly		
Called Procedures	None		

Procedure Name	**Revise**		
Description	To calculates the water requirements with regards of tunnel painting factor and then, update the irrigation schedule.		
Input	*Concept*	*Property*	*Source of value*
	farm	***planting_type***	***database***
	tunnel_painting	***done***	***database***
	irrigation	***Schedule type***	***system***
Output	***Concept***	***Property***	***Source of value***
	irrigation	***wr_after_tunnel_painting_factor***	***function***
	irrigation	***Schedule type***	***system***
Pre-Conditions	The Farm planting_type must be "high_tunnel" and also the tunnel_painting must be "yes"		
Called Procedures	None		
Procedure	For each row in irrigation schedule ***Play Inference*** "tunnel_painting_factor"_r ***Assert*** wr_after_tunnel_painting_factor in the schedule Next i		

Property wr_after_tunnel_painting_factor not found!

Procedure Name	**Output_Irrigation_schedule**		
Description	To display the irrigation schedule according to schedule type and irrigation method.		
Input	*Concept*	*Property*	*Source of value*

	evaluation_result	***value***	***cluster***
	irrigation	***method***	***database***
	irrigation	***Schedule type***	***database***
Output	None		
Pre-Conditions	Evaluation result must be equal “yes”		
Called Procedures	irrigation_units		

Procedure	If Schedule_type = "weekly" and method = "drip" Then
	frmDripIrrDia.Show
	Else
	If Schedule_type = "weekly" and method = "flooding" Then
	frmFloodIrrDia.Show
	Else
	If Schedule_type = "daily" and method = "drip" Then
	frmDailtIrrScr.Show
	Else
	frmIrrExpDlg.Show
	End If
	End If
	End If		

//Concept evaluation result, not found!

Procedure Name	**irrigation_units**		
Description	To update the irrigation schedule with the motors_hours_work and the no_of_irrigation_during_day		
Input	***Concept***	***Property***	***Source of value***
	irrigation	***method***	***database***
	irrigation	***Schedule type***	***system***
Output	***Concept***	***Property***	***Source of value***
	motors_hours_work	***value***	***DG***
	no_of_irrigation_during_day	***value***	***DG***
	irrigation	***Schedule type***	***system***
Pre-Conditions	None		
Called Procedures	None		

Procedure	For each row in the irrigation schedule
	Update irrigation_schedule-water_requirement
	If Method = "drip" Then
	Dependency Graph motors_hours_work
	Dependency Graph no_of_irrigation_during_day
	Update schedule- motors_hours_work
	Update schedule- no_of_irrigation_during_day
	End If
	Next row		

Procedure Name	farmIrrigation_Daily_Data		
Description	This is the main form for the daily irrigation schedule		
Input	***None***		
Output	***Concept***	***Property***	***Source of value***
	irrigation	***last_irr_date***	***user***
	irrigation	***last_wr***	***user***
Pre-Conditions	None		
Called Procedures	None		
Procedure	***Read*** Irrigation Last Date and Last Water Quantity **from the user**		

Procedure Name	irrigation_daily_schedule		
Description	This is the main procedure for the Irrigation Schedule		
Input	***Concept***	***Property***	***Source of value***
	task_parameters	***new_user***	***system***
	task_parameters	***plant_existence***	***database***
	task_parameters	***previous_accumulated_eta***	***system***
	irrigation	***last_wr***	***system***
	irrigation	***controled_water***	***database***
	task_parameters	***days***	***system***
Output	***Concept***	***Property***	***Source of value***
	task_parameters	***previous_accumulated_eta_***	***system***
Output	None		
Pre-Conditions	None		
Called Procedures	- **initialize_irrigation_daily_schedule** - **repeat01** - **while1** - **repeat02**		

Procedure	If (task_parameters-new_user = "yes" And
	task_parameters-plant_existence = "yes") Then
	repeat01
	Else
	ininialize_irrigation_daily_schedule
	End If
	while1(irrigation-controled_water, task_parameters-days)
	repeat02		

//task parameter properties: new_user, plant_existence and days should be defined and explained!

Procedure Name	initialize_irrigation_daily_schedule		
Description	Initialize all facts and variables associated with weekly irrigation schedule task.		
Input	***Concept***	***Property***	***Source of value***
	Current_planting	***date***	***database***
	task_parameters	***today***	***system***
	task_parameters	***plant_existence***	***system***
	farm	***ID***	***database***

	task_parameters	***new_user***	***system***
Output	***Concept***	***Property***	***Source of value***
	task_parameters	**age**	***system***
	task_parameters	*current_date*	***system***
	task_parameters	***no_of_times***	***database***
	task_parameters	*current_month*	***system***
	task_parameters	*current_day*	***system***
	task_parameters	*next_month*	***system***
	task_parameters	***next_irrigation***	***system***
	task_parameters	*interval_number*	***database***
	task_parameters	*previous_accumulated_eta*	***system***
Pre-Conditions	None		
Called Procedures	- **adjusted_irrigation_daily_schedule** - **propose_irrigation_schedule**		

Procedure	If (task_parameters-new_user = "yes") Then
	task_parameters- current_day = task_parameters- today
	task_parameters-no_of_times=0
	task_parameters-next_irrigation=0
	task_parameters-no_interval=0
	task_parameters-current_month= month(task_parameters-current_date)
	If (task_parameters-plant_existence= "no") Then
	task_parameters-previous_accumulated_eta=0
	End If
	Else
	SelectfromDB(Table(irrigation_op<fid, current_planting-date>, values< Acc_eta, No_of_times, No_interval>)
	task_parameters-current_date = current_planting-date +1
	task_parameters-previous_accumulated_eta= Acc_eta
	task_parameters-no_of_times = No_of_times
	task_parameters-no_interval= No_interval
	task_parameters-next_irrigation=0
	task_parameters-current_month = month(task_parameters-current_date)
	End If		

task parameter properties: today, plant_existence, new_user, no_of_times and next_irrigation should be defined and explained

Procedure Name	**repeat01**		
Description	This procedure represents a starting point to determine the irrigation Schedule parameters starting from the current_planting-date up to task_parameters-today		
Input	*Concept*	*Property*	*Source of value*
	Current_planting	*date*	***database***
	task_parameters	***today***	***system***
	task_parameters	*current_date*	***system***
Output	None		

Pre-Conditions	None
Called Procedures	- adjusted_irrigation_daily_schedule - propose_irrigation_schedule

task parameter property: today should be defined and explained

Procedure Name	**while1**		
Description	This procedure represent a starting point to determine the Irrigation Schedule parameters while the irrigation- controled_water is "no" AND task_parameters-days >0		
Input	***Concept***	***Property***	***Source of value***
	irrigation	*controled_water*	*database*
	task_parameters	*days*	*system*
Output	None		
Pre-Conditions	None		
Called Procedures	- adjusted_irrigation_weekly_schedule - propose_irrigation_schedule		

task parameter property: days should be defined and explained

Procedure Name	**repeat02**		
Description	This procedure determine an Irrigation Schedule parameters while the task_parameters-next_irrigation > 0 AND the interval-value = 1		
Input	***Concept***	***Property***	***Source of value***
	interval	*value*	*database*
	task_parameters	*next_irrigation*	*system*
	task_parameters	*current_date*	*system*
	water_requirment	*value*	
Output	***Concept***	***Property***	***Source of value***
	initial_irrigation_schedule	*current_date*	*system*
	initial_irrigation_schedule	*water_requirement*	*system*
	initial_irrigation_schedule	*interval*	*system*
Pre-Conditions	The user must determine the Farm ID and the Schedule type prior to initialize this functions		
Called Procedures	- adjusted_irrigation_daily_schedule - propose_irrigation_schedule		
Procedure	***Dependency Graph*** water_requirement ***Update*** irrigation_schedule- Current_date ***Update*** irrigation_schedule- water_requirement ***Update*** irrigation_schedule- interval		

task parameter property: next_irrigation should be defined and explained and so should concept initial_irrigation_schedule and its properties.

Procedure	**adjusted_irrigation_daily_schedule**

Name			
Description	This procedure determine an Irrigation Schedule parameters while the task_parameters-next_irrigation > 0 AND the interval-value = 1		
Input	***Concept***	***Property***	***Source of value***
	task_parameters	age	*system*
	task_parameters	*current_date*	*system*
	task_parameters	*days*	*system*
Output	***Concept***	***Property***	***Source of value***
	task_parameters	*current_date*	*system*
	task_parameters	age	*system*
	task_parameters	*current_month*	*system*
	task_parameters	*days*	*system*
	accumulated_eta	*value*	*DG*
Pre-Conditions	None		
Called Procedures	- adjusted_irrigation_daily_schedule - propose_irrigation_schedule		
Procedure	***Dependency Graph*** accumulated_eta		

//Why does this procedure call itself?

7 Suggested User interfaces

Below, find a list of suggested user interfaces.

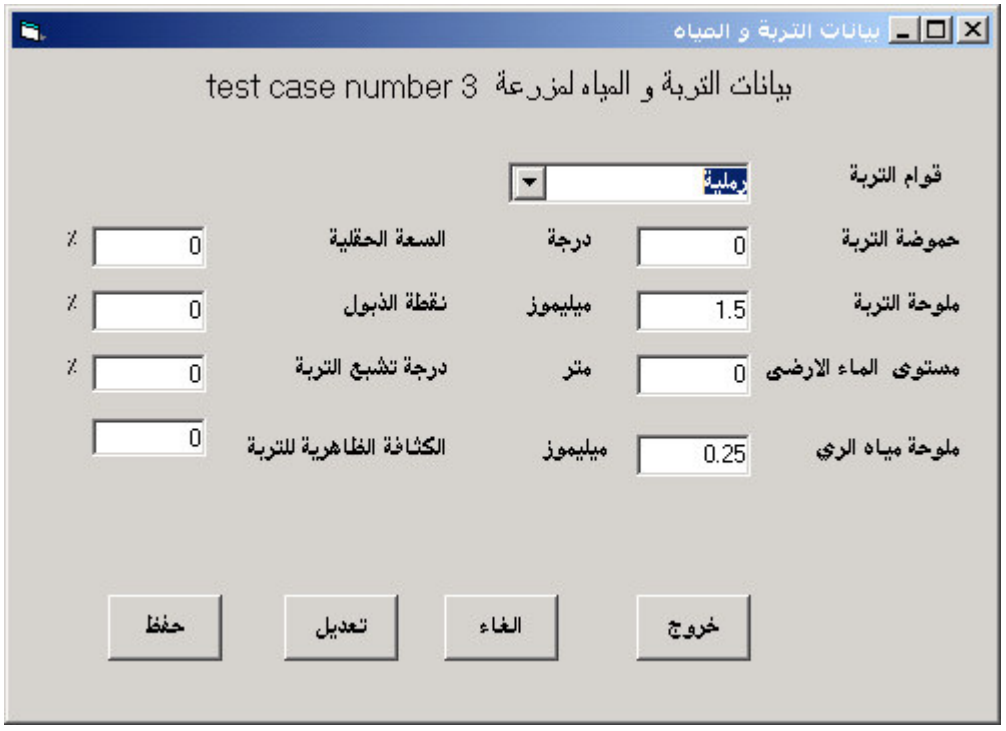

Figure 1: data entry screen for soil and water data

Figure 2: data entry screen for farm data

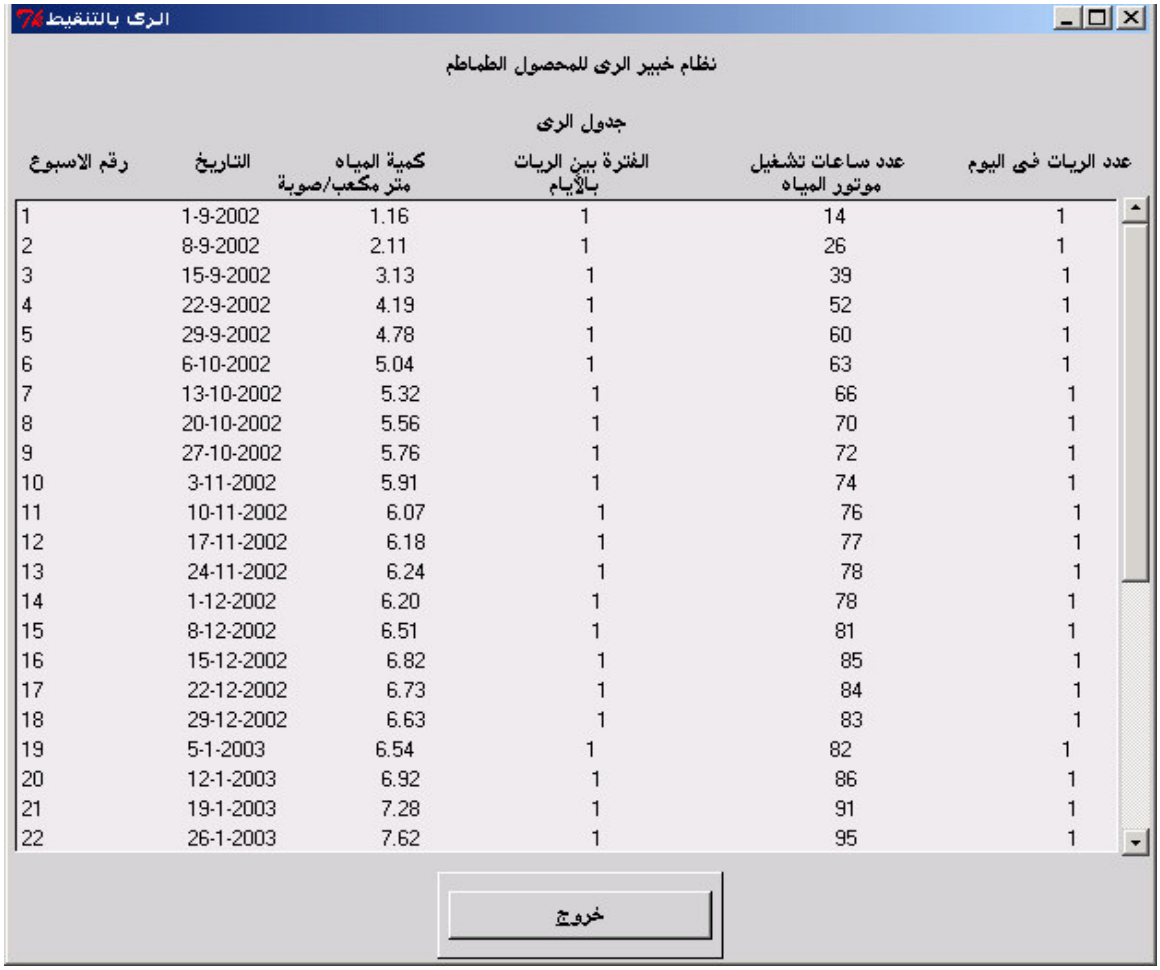

Figure 3: Irrigation output screen for a tunnel