

Strawberry Plant Care Design

1. Introduction

The purpose of this subsystem is to recommend the user about the agricultural operations for the strawberry crop. The system gives either the next agricultural operation or schedule of the agricultural operations for the whole season according to the user requirement. This document consists of 3 parts, domain knowledge, inference knowledge, and task knowledge. CommonKADS methodology is used to represent these knowledge. Sections 2, 3, and 4 present domain, inference, and task knowledge respectively. Section 5 discuss database while section 6 presents the user interface. Section 7 presents the test cases.

2. Domain Knowledge

2.1 Domain Ontology

concept user request;

properties:

value: { }

source of value: user;

cardinality: single;

concept plant;

properties:

status: { }

source of value: user;

cardinality: multiple;

possible status: { }

source of value: derived(PREDICT);

cardinality: multiple;

age: numeric;
source of value: derived;
cardinality: single;

concept plantation;

properties:

area: numeric % in feddan
source of value: database;
cardinality: single;

date: date ;
source of value: user;
cardinality: single;

concept operation;

properties:

method type: {mechanical, manual};
source of value: derived ;
cardinality: single;

tool name: universal;
source of value: derived ;
cardinality: single;

tool hiring rate: numeric;
source of value: database;
cardinality: single;

total time: numeric % hour per feddan
source of value: derived ;
cardinality: single;

labor age: {men, boys};
source of value: derived ;
cardinality: single;

number of boys per feddan: numeric;
source of value: derived ;
cardinality: single;

number of men per feddan: numeric;
source of value: derived ;
cardinality: single;

status: {suggested, not suggested, cancelled};
source of value: derived (SUGGEST);
cardinality: single;

default value: not suggested

occurrence: { };
source of value: user;
cardinality: single;
default value:

importance: { };
source of value: derived;
cardinality: single;

method: universal;
source of value: derived(ASSIGN);
cardinality: single;

operation number: numeric;
source of value: derived(APPLY);
cardinality: single;

cost: numeric;
source of value: derived;
cardinality: single;
default: 0

actual cost: numeric;
source of value: user;
cardinality: single;
default: 0

name: universal;
source of value: derived(APPLY);
cardinality: multiple;

concept operation cost;

properties:

value: { , };
source of value: user ;
cardinality: single;

concept seedling;

properties:

sub-type-of: plant;

type: { };
source of value: database;
cardinality: single;

concept session date;

properties:

current month: numeric;
source of value: system;
cardinality: single;

value: date; /* system date */
source of value: system;
cardinality: single;

concept Irrigation;

properties:

type: { };
source of value: database;
cardinality: single;

concept Planting;

properties:

type: { };
source of value: database;
cardinality: single;

concept event;

properties:

occurrence: { };
source of value: user;
cardinality: single;
default value:

value: { };
source of value: user;
cardinality: multiple ;

concept soil;

properties:

type: { };
source of value: database;
cardinality: single;

concept previous crop;

properties:

fertilizer: { }
source of value: database;
cardinality: single;

concept ;

sub-type-of: operation;
importance: ;

tool name:
method type: mechanical
total time: 4

concept ;
sub-type-of: operation;
importance: ;

tool name:
method type: mechanical
total time: 4

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 4

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 1

concept ;
sub-type-of: operation;
importance: ;
tool name:
method type: mechanical
total time: 4

concept ;
sub-type-of: operation;
importance: ;
tool name: { , }
 source of value: database;
 cardinality: single;
method type: mechanical;

total time: 4

concept ;

sub-type-of: operation;

importance: ;

tool name: +

method type: mechanical

total time: 4

concept ;

sub-type-of: operation;

importance: ;

method type: manual

labor age: men

number of men per feddan: 3

concept ;

sub-type-of: operation;

importance: ;

tool name: +

method type: mechanical

total time: 2

concept ;

sub-type-of: operation;

importance: ;

method type: manual

labor age: men

number of men per feddan: 4

concept ;

sub-type-of: operation;

importance: ;

method type: manual

labor age: men

number of men per feddan: 3

concept ;

sub-type-of: operation;

importance: ;

tool name:
method type: mechanical
total time: 4

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 6

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 6

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 2

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 4

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 4

concept ;
sub-type-of: operation;
importance: ;
method type: manual

labor age: boys
number of boys per feddan: 6

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: boys
number of boys per feddan: 6

concept ;
sub-type-of: operation;
importance: ;
application date: date
 source of value: derived(SUGGEST);
 cardinality: single;
method type: manual
labor age: men
number of men per feddan: 4

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: boys
number of boys per feddan: 6

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: men
number of men per feddan: 6

concept ;
sub-type-of: operation;
importance: ;
method type: manual
labor age: boys
number of boys per feddan: 15

concept ;
sub-type-of: event;

concept ;
sub-type-of: event;

concept ;
sub-type-of: event;

relation: SUGGEST;
argument-1: plantation, event, operation, plant, pods, soil;
argument-role: environment;
argument-2: operation;
argument-role: status of next operation;

relation: ASSIGN;
argument-1: plantation, operation, soil;
argument-role: observations;
argument-2: operation;
argument-role: operation method;

relation: PREDICT;
argument-1: plant;
argument-role: plant age;
argument-2: plant;
argument-role: possible plant status;

relation: APPLY;
argument-1: plantation, plant, irrigation, seedling, soil;
argument-role: environment;
argument-2: operation;
argument-role: agricultural operations;

2.2 Domain Models

domain-model: suggestion model;
parts: tuple(suggest);

axioms:

seedling : type = &
(session date : current month = 9 OR
session date : current month = 10)
: occurrence =
SUGGEST
(: status = suggested)

seedling : type = &

```

(session date : current month = 8 OR
 session date : current month = 9)
    : occurrence =
      SUGGEST
    (
      : status = suggested )
      occurrence = &
    : occurrence =
      SUGGEST
    (
      : status = suggested )
    (
      : occurrence = &
      : occurrence = &
    (soil: type = OR
 previous crop : fertilizer = ))
      SUGGEST
    (
      : status = suggested )
      : occurrence = &
    ((soil: type = OR : type = ) &
 previous crop : fertilizer = )
      SUGGEST
    (
      : status = cancelled)
      : occurrence = &
    : occurrence = &
    (
      : occurrence = OR
      : occurrence = )
      SUGGEST
    (
      : status = suggested )
      : occurrence = &
    (
      : status = cancelled &
      : occurrence = )
      SUGGEST
    (
      : status = suggested )
      : occurrence = &

```

: occurrence =
SUGGEST
(: status = suggested)

: occurrence = &
Irrigation: type = &
: occurrence =
SUGGEST
(: status = suggested)

: occurrence = &
Irrigation: type = &
: occurrence =
SUGGEST
(: status = cancelled)

: occurrence = &
(: occurrence = OR
: occurrence = OR
: status = cancelled)
SUGGEST
(: status = suggested)

: occurrence = &
Irrigation: type = &
: occurrence =
SUGGEST
(: status = suggested)

: occurrence = &
Irrigation: type = &
: occurrence =
SUGGEST
(: status = suggested)

: occurrence = &
Irrigation: type = &

: occurrence =
 SUGGEST
 (: status = suggested)

 : occurrence = &
 : occurrence = &
 Irrigation: type =
 SUGGEST
 (: status = suggested)

 : occurrence = &
 : occurrence = &
 Planting: type =
 SUGGEST
 (: status = suggested)

 : occurrence = &
 Irrigation: type = &
 : occurrence =
 SUGGEST
 (: status = suggested)

 : occurrence = &
 Irrigation: type = &
 : occurrence =
 SUGGEST
 (: status = suggested)

 : occurrence = &
 : occurrence = &
 seedling: type = &
 (session date : current month = 9 OR
 session date : current month = 10)
 SUGGEST
 (: status = suggested)

 : occurrence = &
 Irrigation: type = &

: occurrence = &
(session date : current month = 8 OR
session date : current month = 9)
SUGGEST
(: status = suggested)

: occurrence = &
Irrigation: type = &
seedling : type = &
: occurrence = &
(session date : current month = 8 OR
session date : current month = 9)
SUGGEST
(: status = suggested)

: occurrence = &
: occurrence = &
Irrigation: type = &
Planting: type = &
seedling : type =
SUGGEST
(: status = suggested)

seedling : type = &
: occurrence = &
Plant: age < 25 &
Plant: age >= 15 &
Plant: status =
SUGGEST
(: status = suggested)

seedling : type = &
: occurrence = &
Plant: age < 25 &
Plant: age >= 15 &
Plant: status =
SUGGEST
(: status = suggested)

seedling : type = &
: occurrence = &
Plant: age < 60 &
Plant: age >= 15 &
Plant: status = &
SUGGEST
(: status = suggested)

seedling : type = &
: occurrence = &
Plant: age < 180 &
Plant: age >= 15 &
Plant: status = &
SUGGEST
(: status = suggested)

Irrigation: type = &
Planting: type = &
seedling : type = &
: occurrence = &
: occurrence = &
: occurrence = &
: occurrence =
SUGGEST
(: status = suggested)

Irrigation: type = &
Planting: type = &
seedling : type = &
: occurrence = &
: occurrence = &
(: occurrence = OR
: occurrence =)
SUGGEST
(: status = cancelled)

seedling : type = &
: occurrence =
plant: age > 10 &
plant: age < 90 &
plant: status =
SUGGEST
(: status = suggested)

seedling : type = &
: occurrence =
plant: age >= 30 &
plant: age < 90 &
plant: status =
SUGGEST
(: status = suggested)

: occurrence = &
seedling : type = &
Plant: age > 60 &
plant : status =
SUGGEST
(: status = suggested)

: occurrence = &
seedling : type = &
Plant: age >180 &
plant : status =
SUGGEST
(: status = suggested)

: occurrence = &
Plant: age >= 30 &
Plant: age <= 60
SUGGEST
(: status = suggested &
: application date = session date: value)

: occurrence = &
Plant: age <= 60 &

Session date: value - : application date >= 10

SUGGEST

(: status = suggested &

: application date = session date: value)

: occurrence = &

: occurrence = &

session date : current month = 4 &

: occurrence =

SUGGEST

(: status = suggested)

seedling: type = &

Plant: age > 60 &

Plant: age < 250 &

Plant: status =

SUGGEST

(: status = suggested)

seedling: type = &

Plant: age >180 &

Plant: age < 340 &

Plant: status =

SUGGEST

(: status = suggested)

domain-model: application model;

part: tuple(apply);

axioms:

seedling : type = OR

seedling : type =

APPLY

Operation: name = &

: operation number = 1

seedling : type = OR

seedling : type =

APPLY

Operation: name = &
 : operation number = 2

(seedling : type = OR
 seedling : type =) &

(soil: type = OR
 previous crop : fertilizer =)

APPLY

Operation: name = &
 : operation number = 3

seedling : type = OR
 seedling : type =

APPLY

Operation: name = &
 : operation number = 4

seedling : type = OR
 seedling : type =

APPLY

Operation: name = &
 : operation number = 5

Irrigation: type = &
 seedling : type = &

APPLY

Operation: name = &
 : operation number = 6

seedling : type = OR
 seedling : type =

APPLY

Operation: name = &
 : operation number = 7

Irrigation: type = &

seedling : type =
SUGGEST
Operation: name = &
: operation number = 8

seedling : type = OR
seedling : type =
APPLY
Operation: name = &
: operation number = 9

Irrigation: type =
APPLY
Operation: name = &
: operation number = 10

Planting: type =
APPLY
Operation: name = &
: operation number = 11

Irrigation: type = OR
Irrigation: type = &
APPLY
Operation: name = &
: operation number = 12

seedling : type = OR
seedling : type =
APPLY
Operation: name = &
: operation number = 13

Irrigation: type = &
Planting: type = &
seedling : type =
APPLY

Operation: name = &
: operation number = 14

seedling : type =
APPLY
Operation: name = &
: operation number = 15

seedling : type = OR
seedling : type =
APPLY
Operation: name = &
: operation number = 17

seedling : type = OR
seedling : type =
APPLY
Operation: name = &
: operation number = 16

seedling : type =
APPLY
Operation: name = &
: operation number = 18

seedling : type =
APPLY
Operation: name = &
: operation number = 19

seedling : type = OR
seedling : type =
APPLY
Operation: name = &
: operation number = 20

seedling : type = OR

seedling : type =
APPLY
Operation: name = &
: operation number = 21

seedling : type =
APPLY
Operation: name = &
: operation number = 22

seedling : type = OR
seedling : type =
APPLY
Operation: name = &
: operation number = 23

domain-model: assignment model;
part: tuple(assign);

axioms:

: status = suggested &
ASSIGN
: method =

: status = suggested &
ASSIGN
: method = 30 - 25

: status = suggested &
ASSIGN
: method = 30 - 25

: status = suggested
ASSIGN

```

: method =

: status = suggested &
ASSIGN
: method =
"

: status = suggested &
ASSIGN
: method = "
"
&

: status = suggested &
seedling : type =
ASSIGN
: method = " 120 -100 "

: status = suggested &
Irrigation: type = &
seedling : type = &
Planting: type = &
ASSIGN
: method = " 120 -100 "

: status = suggested &
seedling : type = &
Planting: type = &
ASSIGN
: method = " 10 -9 "

: status = suggested
ASSIGN
: method = - 7
"
"
10

```

: status = suggested &
Irrigation: type =
ASSIGN
: method = "

: status = suggested &
Irrigation: type =
ASSIGN
: method = - - "

: status = suggested &
Irrigation: type =
ASSIGN
: method = " () "

: status = suggested &
Irrigation: type =
ASSIGN
: method = " 2-1 "

: status = suggested
ASSIGN
: method = "
" / 4 25 "

: status = suggested
ASSIGN
) ": method =
(
20 20
" 20 - 15 U 20

: status = suggested &

```

seedling : type =
    ASSIGN
    2-1          ": method =
15             (      40 -   35   )
    20 - 15
                "
                : status = suggested &
Planting: type =          &
seedling : type =
    ASSIGN
    : method =   20 -   16   )      "
                15             (
"                30 - 25
                : status = suggested &
Planting: type =          &
seedling : type =
    ASSIGN
    : method =   20 -   16   )      "
25             15             (
"                30 -
                : status = suggested
    ASSIGN
    : method =           2 -1.5      "
"                50
                : status = suggested
    ASSIGN
    : method =          "
"
                : status = suggested &

```

Planting: type =
ASSIGN
: method = " "

: status = suggested &
Planting: type = &
ASSIGN
: method = " "

: status = suggested &
ASSIGN
: method =

: status = suggested
ASSIGN
: method = " "

: status = suggested
ASSIGN
: method = " "

: status = suggested
ASSIGN
: method = " "

: status = suggested
ASSIGN
: method = " "

: status = suggested
ASSIGN


```

: method = 250 "
"
100
: status = suggested &
seedling: type = &
ASSIGN
: method = "
"
16 - 14
: status = suggested &
seedling: type = &
ASSIGN
: method = "
"
12 - 8

```

domain-model: prediction model;
parts: tuple(predict)

axioms:

```

seedling : type = &
: occurrence =
plant: age > 10 &
plant: age < 30
PREDICT
(plant: possible status = [ ])

seedling : type = &
: occurrence =
plant: age >= 30 &
plant: age < 90 &
PREDICT
(plant: possible status = [ ])

seedling : type = &
Plant: age > 60 &
PREDICT
(plant: possible status = [ ])

```

seedling : type = &
Plant: age >180 &
PREDICT
(plant: possible status = [])

Plant: age < 25 &
Plant: age >= 15 &
PREDICT
(plant: possible status = [])

seedling : type = &
Plant: age < 60 &
Plant: age >= 25 &
PREDICT
(plant: possible status = [])

seedling : type = &
Plant: age < 180 &
Plant: age >= 25 &
PREDICT
(plant: possible status = [])

3. Inference Knowledge

3.1 Inference Structure

Inference structure is shown in figure 1.

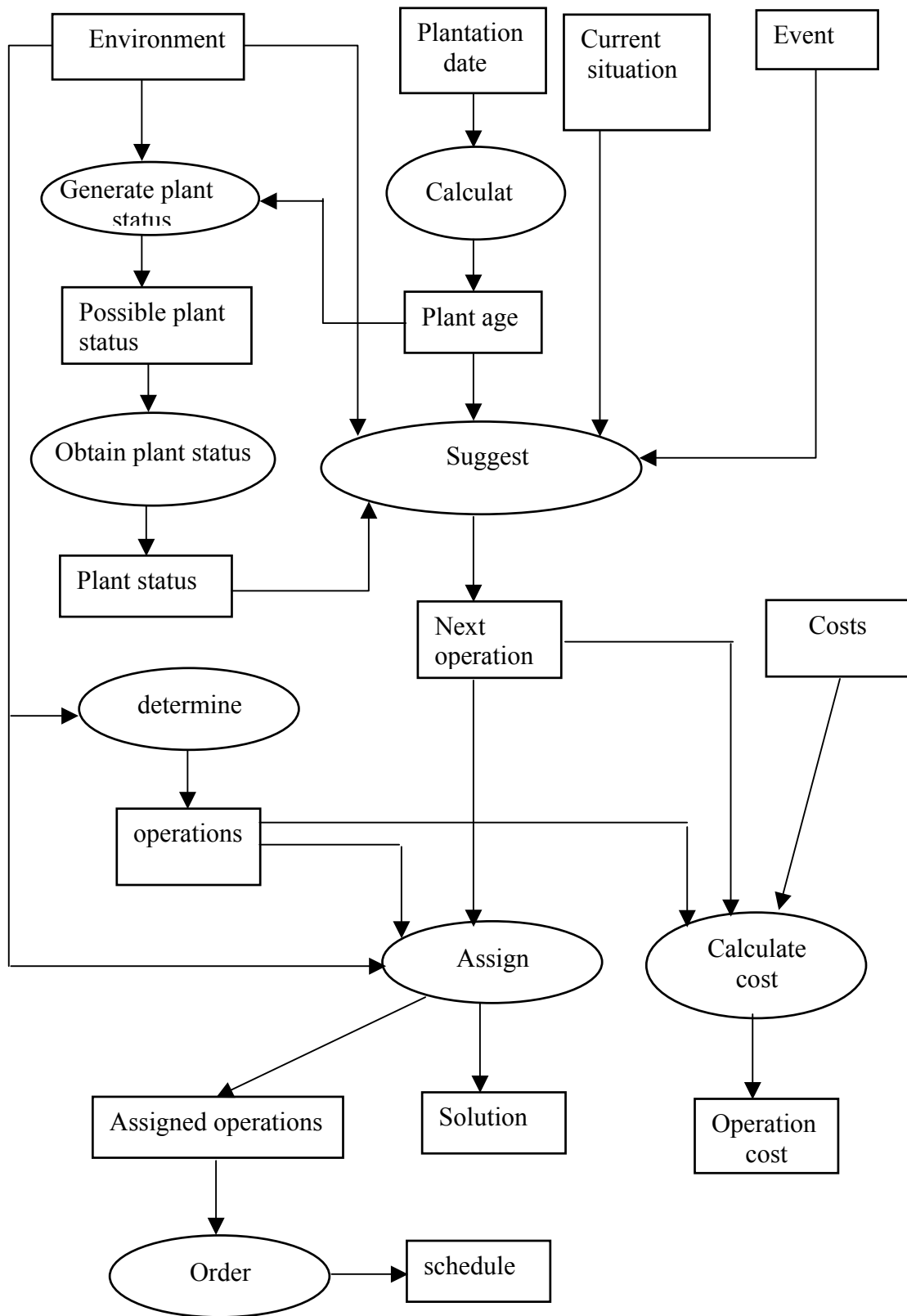


Fig. (1): inference structure

3.2 Inference specification

inference: calculate

operation-type: calculate the plant age.

input-roles: plantation date.

output-roles: plant age.

static-roles: there is no static roles

spec: plant age is calculated in days by subtracting the plantation date from the session date.

inference: generate plant status

operation-type: generates the possible plant status.

input-roles: environment, plant age.

output-roles: possible plant status.

static-roles: PREDICT \in prediction-model.

spec: the possible plant status are generated by applying "PREDICT " relation.

inference: obtain plant status

operation-type: transfer task.

input-roles: possible plant status.

output-roles: plant status.

static-roles: there is no static roles.

spec: obtain the current plant status from the user.

inference: suggest

operation-type: suggest the next agricultural operation.

input-roles: environment, plant age, event, plant status, current situation.

output-roles: next operation.

static-roles: SUGGEST \in suggestion-model.

spec: the next agricultural operation are to be suggested by applying " SUGGEST " relation.

inference: determine

operation-type: determine the agricultural operations schedule.

input-roles: environment.

output-roles: operations.

static-roles: APPLY \in application model.

spec: the agricultural operations schedule is determined by applying " APPLY" relation.

inference: assign

operation-type: assign parameters to the suggested operation.

input-roles: next operation, environment, operations.

output-roles: solution, assigned operations.

static-roles: ASSIGN \in assignment-model.

spec: assign the method to the suggested operation or to the scheduled operations by applying " ASSIGN " relation.

inference: order

operation-type: order the operations ascending according to
operation: operation number .

input-roles: assigned operations

output-roles:. schedule

static-roles: there is no static role

spec: get operation: name(L), /* L is a list */
for each element(E) in L get E: operation number,
sort ascending L according to operation number of each element,
the sorted list is L',
DISPLAY(L')

inference: calculate cost

operation-type: calculate cost of the suggested operation.

input-roles: next operation, costs.

output-roles: operation cost.

static-roles: there is no static role

spec:

```
IF the user request is (next operation)
THEN
  Begin
    IF the method of the next operation is mechanical
    THEN  get the tool name (T) of the next operation
          get the tool hiring rate (THR) of (T)
          get the total time/Feddan (TT) of applying the next operation
          get the area (A) of the plantation
          cost = THR*TT*A
    IF the method of the next operation is manual
    THEN  get the number of men/Feddan (M) of applying the next
operation
          get the labor wage (W)
          get the area (A) of the plantation
          cost = M*W*A
    End
  ELSE
  Begin
    get operation: name (N)          % N is a list
    While N <> {}
    Begin
      N = [E | Tail]
      IF the method of 'E' is mechanical
      THEN  get the tool name (T) of 'E'
            get the hiring rate (HR) of (T)
            get the total time/Feddan (TT) of applying 'E'
            get the area (A) of the plantation
            Cost = HR*TT*A
            Insert Cost in E: cost
            TotalCost = TotalCost + Cost
            N = Tail
```

```

IF the method of 'E' is manual
THEN
    IF E: labor age = men
    THEN
        get number of men per feddan( (No) of 'E'
        get man wage ( ) (W)
    IF E: labor age = boys
    THEN
        get number of boys per feddan( (No) of 'E'
        get boy wage ( ) (W)

        get the area (A) of the plantation
        Cost = No*W*A
        Insert Cost in E: cost
        TotalCost = TotalCost + Cost
        N = Tail
End

```

4. Task Knowledge

task: strawberry plant care;

task-definition:

goal: suggest the next agricultural operation,

presents a schedule of the agricultural operations;

input: Environment: {soil: type, seedling : type, previous crop : fertilizer,
Irrigation: type, Planting: type };

plant status: { plant: status};

plantation date: {plantation: date};

current situation: {operation: occurrence};

event: {event: value};

cost: {the cost of equipments, labor , ... etc from database}

output: solution: { suggested operation, importance, method},

operation cost: {cost of the suggested operation},

schedule: {the schedule of the agricultural operation during the season},

task-body:

type: composite

subtasks: calculate, generate plant status, obtain plant status, suggest, assign,
determine, order

additional-roles:

Possible plant status: {plant: possible status }

Plant status {plant: status}

plant age: {plant: age}

Next operation {operation: status = suggested}

Operations {operation: name}

control-structure:

IF the user request: value =

```

THEN
BEGIN
    IF seedling: type = &
        (session date: current month < 9 OR
        session date: current month > 10) &
            : occurrence =
    THEN prompt the user " "
    ELSE
    IF seedling: type = &
        (session date: current month < 8 OR
        session date: current month > 9) &
            : occurrence =
    THEN prompt the user " "
    ELSE

        OBTAIN (last suggested operation (OP)) % from database,
        OBTAIN (OP: occurrence) % from user
        Assert (OP: occurrence) % in database
        IF (OP: occurrence = OR ) & OP: importance =
        THEN BEGIN
            prompt the user
            " " "OP " "
        END
    ELSE BEGIN
        IF (OP: occurrence = ) & operation cost: value =
        THEN prompt the user to type the actual cost of OP
            Assert(OP: actual cost) % in database
        IF ( : occurrence = )
        THEN BEGIN
            IF plantation: date is UNKNOWN
            THEN BEGIN
                OBTAIN (plantation: date), % from the user
                calculate(PD: plantation: date → PA: plant: age)
                IF plant: age > 10
                THEN
                    generate plant status(E: environment →
                    P: possible plant status),
                    obtain plant status(P → PS: plant status)
                END
            IF seedling : type = & : occurrence =

```

```

        THEN BEGIN OBTAIN (event: occurrence) END
        END
suggest(E, CS: current situation, PA, EV: event, PS →
        NO: next operation),
IF NO is empty
THEN
    Prompt the user ‘
ELSE
    BEGIN
    assert(NO) in database,
    assign(NO, E → Sol: solution),
    IF operation cost: value =
    THEN calculate cost(NO, C: costs → OC: operation cost)
    PRESENT(Sol, OC)
    ELSE
    PRESENT(Sol)

    END
END
END
ELSE IF the user request: value =
    THEN determine(E: environment → O: operations),
    assign(E, O → AP: assigned operations),
    order(AP → S: schedule),
    IF operation cost: value =
    THEN calculate cost(O, C: costs → OC: operation cost)
    PRESENT(S, OC)
    ELSE PRESENT(S)

```

5. Database

The data base of strawberry plant care is exactly similar to database of bean after updating the legal values and introducing the following comments:

- Delete plantation date from the database interface since it is obtained from the user during run time and inserted in database.
- Delete crop type, seed status and plantation type from the database.
- Add the following items:

(seedling: type { })

(irrigation: type {

})

(planting: type {

})

(: tool name { , })

- Add the tool hiring rate for the following tools:

% per hour

% per hour

+

% per hour

% per hour

% per hour

- Restrictions on values:

seedling: type = → irrigation: type = &
planting: type =

planting: type = → irrigation: type =

irrigation: type = → planting: type =

soil: type = → irrigation: type =

soil: type = → irrigation: type =

6. User Interface

6.1. Input

The input screen of strawberry plant care is similar exactly as the input screen of bean plant care except that the menu ‘ ’ which is the legal value of ‘event: value’ has

to be activated only if 'seedling : type = ' and ' : occurrence = ' .
Add a button ' ' and another button ' '. The first button
assert ' ' to the property 'user request: value' while the second
button assert the value ' '. Add pull down menu of ' ' that has two
values ' , ' , the selected value is to be added in the property 'operation
cost: value'

6.2. Output

In the cases of the user requirement is ' ' the output screen of
strawberry plant care is similar exactly as the output screen of bean plant care except
that the cost of the suggested operation is added. In the case of the user requirement is
' ' the output screen is about a list of the agricultural operations obtained
from the system.

7. Test Cases

Case 1 (new plantation)

Input

Session date: 1/2/2002
user request:
seedling type:
soil type:
irrigation type:
planting type:

Output

Case 2

Input

Session date: 15/9/2002

user request:

seedling type:

soil type:

irrigation type:

planting type:

previous crop : fertilizer =

Output

:
:
:

Case 3

Input

Session date: 18/9/2002

user request:

: occurrence =

Output

:
:

30 - 25

:

Case 4

Input

Session date: 19/9/2002

user request:

: occurrence =

Output

:

:

:

Case 5

Input

Session date: 19/9/2002

user request:

: occurrence =

Output

:

:

:

Case 6

Input

Session date: 20/9/2002

user request:

: occurrence =

Output

:

:

30 - 25

:

Case 7

Input

Session date: 20/9/2002

user request:

: occurrence =

Output

:

120 -100

"

:
:

Case 8

Input

Session date: 21/9/2002

user request:

: occurrence =

Output

:

:
:

Case 9

Input

Session date: 21/9/2002

user request:

: occurrence =

Output

:

25
" / 4

:
:

Case 10

Input

Session date: 21/9/2002
user request:
: occurrence =

Output

20 ()
20 20 20
20 – 15 Case 3
U

:
:
:

Case 11

Input

Session date: 22/9/2002
user request:
: occurrence =

Output

:

35) 2-1
 15 (40 -
 20 - 15
 "

Case 12

Input

Session date: 22/9/2002
 user request:
 : occurrence =

Output

50 2 -1.5 :
 "

Case 13

Input

Session date: 23/9/2002
 user request:
 :
 plantation date: 22/9/2002

Output

"

:
:
:

Case 14

Input

Session date: 24/9/2002
user request:
:

Output

:
:
:

Case 15

Input

Session date: 25/9/2002
user request:

Output

:
:

:

"

Case 16

Input

Session date: 6/10/2002

user request:

Plant status: &

Output

:

:

:

"

:

:

"

"

:

:

" :

"

Case 17

Input

Session date: 7/10/2002
user request:

Output

"
:
:
:"

Case 18

Input

Session date: 20/10/2002
user request:
Plant status:

Output

"
:
:
:"

"
:
:
:"

Case 19

Input

Session date: 22/10/2002
user request:

Output

" :
 :
 :
 :
 :
 100 250 " :
 "

Case 20

Input

Session date: 25/10/2002
user request:

:

Output

:

:
:

"

Case 21

Input

Session date: 26/10/2002

user request:

:

Output

:

:

:

Case 22

Input

Session date: 2/11/2002

user request:

Output

:

:

"

:

:

:

100

250

":

"

Case 23

Input

Session date: 10/11/2002

user request:

Output

:

:

:

"

Case 24

Input

Session date: 12/11/2002

user request:

Output

:

:

:

"

:

:

100

250

":

"

Case 25

Input

Session date: 23/11/2002

user request:

plant status:

:

Output

:

:

" " :

Case 26

Input

Session date: 25/11/2002

user request:

Output

:

:

:

"

Case 27

Input

Session date: 26/11/2002

user request:

Plant: status =

Output

:

:

:

"

:

:

"

" 16 – 14

Case 28

Input

Session date: 2/4/2003

user request:

:

Output

:
:
:
:
:
:"

Case 29

Input

Session date: 4/4/2003
user request:
Plant: status =

Output

:
:
:"

" 16 - 14

Case 30 (new plantation)

Input

user request:

seedling type:
soil type:
irrigation type:
planting type:
previous crop : fertilizer =

Output

Case 31 (new plantation)

Input

user request:
seedling type:
soil type:
irrigation type:
planting type:
previous crop : fertilizer =

Output

APPENDIX I

Agricultural Operations Knowledge

:

:

:

:

:

:

1 :

1 :

:

:

:

:

4-3 :

:

:

:

:

:

:

:

:

:

:

1 :

2 :

:

:

:

5-1 :

/ / 4 :

:

:

30 - 25

:

:

12 :
20 :
4 :
3 :
1 :
/ / 4 :
20 :
:

:

:

:

:

:

:

1 :

4 :

:

:

:

:

/ / 1 :

:

:

:

:

:

:

:

:

:

:

1 :

5 :

:

:

:

:

/ / 4 :

:

:

30 - 25

:

:

:

:

:

:

:

:

1 :

6 :

:

:

:

:

/ / 4 :

:

:

:

:

:

:

:

:

:

:

1 :

7 :

:

:

:

:

/ / 4 :

:

:

:

:

-

10 - 9

120 -100

()

:

-

10 -9

+

:

:

:

:

:

:

:

1 :

8 :

:

:

:

:

/ . 4 / / / 2 :

:

:

:

10 - 7

+

:

:

:

:

:

:

:

1 :

9 :

:

:

:

:

/ . 3 :

:

:

:

:

-

:

-

-

-

-

:

:

:

:

:

:

:

1 :

10 :

:

:

:

:

/ . 4 :

18 16

:

/ 4

25

:

:

" / 4

25

:

:

:

:

:

:

:

1 :

11 :

:

:

:

:

/ . 4 :

:

:

:

20

20

"

20

" 20 - 15

U

:

:

:

:

:

:

:

1 :

12 :

:

:

:

:

/ 4 :

:

:

:

:

-

2-1

:

-

()

:

:

:

:

:

:

:

1 :

13 :

:

:

:

:

/ . 6 :

/ 40 - 35 :

:

:

:

-

15

"

20 - 15

"

:

-

15

"

"

20 - 15

:

:

:

:

:

:

:

1 :

13 :

:

:

:

:

/ . 6 :

/ 20 - 16 :

:

:

:

-

15

"

30 - 25

"

:

-

15

"

"

30 - 25

:

:

:

:

:

:

:

1 :

14 :

:

:

:

:

/ . 6 :

/ 300 50 :

:

" 50 2 -1.5 "

:

:

:

:

:

:

:

1 :

15 :

:

() :

20 - 15 :

:

/ . 2 :

:

:

"

:

:

:

:

:

:

:

3 -2 :

16 :

:

:

5 :

:

/ . 4 :

:

:

:

-

"

"

:

-

"

"

:

:

:

:

:

:

:

:

17 :

:

:

:

:

/ . 4 :

:

:

:

"

:

:

:

:

:

:

:

:

18 :

:

:

:

:

/ . 6 :

:

:

:

"

:

:

:

:

:

:

:

:

19 :

:

:

:

:

/ . 6 :

:

:

:

:

:

:

:

:

:

:

4 - 3 :

20 :

:

:

:

:

/ . 6 :

:

:

:

"

:

:

:

:

:

:

:

4 - 3 :

20 :

:

:

6 :

:

/ . 6 :

:

:

:

"

:

:

::

:

:

:

:

4 :

21 :

:

:

:

:

/ . 4 :

/ 300/ 750 :

:

300 :

750

:

:

::

:

:

:

:

1 :

22 :

:

:

:

:

/ . 6 :

:

:

:

"

:

:

::

:

:

:

:

:

23 :

:

:

:

:

/ / . 15-10 :

:

:

:

"

" 16 - 14

:

:

::

:

:

:

:

:

23 :

:

:

6 :

:

/ / . 15-10 :

:

:

:

- 8

" 12

: