

APPENDIX

Appendix 1. Source Codes of Limbukha Model

1. step: t

“Simulation”

```
self theFarmers do:
    [:a | a release.
    a decideCropSuccession].
self theMarkets first init.
self theRains first init.
(self theFarmers select: [:a | a myCropSuccession name = 'potatoRice'])
    do: [:b | b plantPotato].
self theVillages do: [:a | a updateWaterShare].
self theFarmers do: [:a | a calculateWaterLabor].
(self theFarmers select: [:a | a waterToBeExchanged > 0]) do:
    [:b | b kinship notEmpty
    ifTrue:
        {self halt.
        b sendInKinship}].
self theFarmers do: [:a | a consultMailBox].
self theFarmers do:
    [:b |
    b definePeopleToAsk.
    b exchangeWater].
[(self theFarmers
    select: [:a | a waterToBeExchanged < 0 and: {a peopleToAsk isEmpty not}]]
    size > 0)
    whileTrue: {self theFarmers do: [:a | a exchangeWater]}.
self theFarmers do: [:a | a plantRice].
self halt.
self theFarmers do: [:a | a harvestPotato].
self theRains first init.
self theVillages do: [:a | a updateWaterShare].
self theFarmers do: [:a | a calculateWaterSecondCycle].
(self theFarmers select: [:a | a waterToBeExchanged > 0])
    do: [:b | b kinship notEmpty ifTrue: [b sendInKinship]].
self theFarmers do: [:a | a consultMailBox].
self theFarmers do:
    [:b |
    b definePeopleToAsk.
    b exchangeWater].

[(self theFarmers
    select: [:a | a waterToBeExchanged < 0 and: {a peopleToAsk isEmpty not}]]
```

```

        size > 0]
        whileTrue: [self theFarmers do: [:a | a exchangeWater]].
self theFarmers do: [:a | a plantRice].
self halt.
self theFarmers do:
    [:a |
    a harvestRice.
    a sellProduction]

```

2. decideCroppingPattern

“Decide the crop sequence based on rainfall pattern and market state”

```

self myCroppingPattern: CroppingPattern new.
(self myVillage id = 1 and:
    {self myMarket marketState = #high
    and: [self myVillage myRain rainState = #high]})
    ifTrue: [self myCroppingPattern name: 'potatoRice']
    ifFalse: [self myCroppingPattern name: 'rice'].
self defineVisualState decideCroppingPattern
self myCroppingPattern: CroppingPattern new.
(self myVillage id = 1 and:
    {self myMarket marketState = #high
    and: [self myVillage myRain rainState = #high]})
    ifTrue: [self myCroppingPattern name: 'potatoRice']
    ifFalse: [self myCroppingPattern name: 'rice'].
self defineVisualState

```

3. plantPotato

“Plant potato and update the income”

```

| counter c |
counter := 0.
self myField components do: [:a | counter < 3
    ifTrue:
        [c := Crop new.
        c cropType: 'potato'.
        c isMovedTo: a.
        counter := counter + 1.
        self myIncome: self myIncome - 5200]]

```

4. calculateWaterLabor

“Calculate number of labor and water units available for sharing”

```

self laborToBeExchanged: self myLabor.
self waterToBeExchanged: self myWater.

```

```

self myField components do: [:a | a crop isEmpty
    ifTrue:
        [self laborToBeExchanged: self laborToBeExchanged - 20.
        self waterToBeExchanged: self waterToBeExchanged - 1]].
self laborExchanged: 0.
self waterExchanged: 0

```

5. plantRice

“Plant rice and update income”

```
| c w l |
```

```

w := self myWater + self waterExchanged.
l := self myLabor + self laborExchanged.
self myField components do:
    [:a |
        (a crop isEmpty and: [w > 0 and: [l >= 20]])
        ifTrue:
            [c := Crop new.
            c cropType: 'rice'.
            c isMovedTo: a.
            w := w - 1.
            l := l - 20.
            self myIncome: self myIncome - 2300]]

```

6. harvestPotato

“Harvest potato and update potato production”

```

| yield p c |
self myVillage myRain rainState = #high ifTrue: [yield := 2200].
self myVillage myRain rainState = #low ifTrue: [yield := 700].
p := self myField components select: [:a | a crop isEmpty not and: [a crop first
cropType = 'potato']]. p
do:
    [:a |
        c := a crop first.
        c leave.
        self myPotatoProduction: self myPotatoProduction + yield]

```

7. harvestRice

“Depending on the rainstate inform the yield. harvest rice and update rice production”

```

| yield p c |
self myVillage myRain rainState = #high ifTrue: [yield := 600].
self myVillage myRain rainState = #low ifTrue: [yield := 400].

```

```
p := self myField components select: [:a | a crop isEmpty not and: [a crop first
cropType = 'rice']].
```

```
p do:
```

```
[:a |
c := a crop first.
c leave.
self myRiceProduction: self myRiceProduction + yield]
```

8. sellProduction

“Sell potato, rice and update income”

```
self
myIncome: self myIncome
+ (self myPotatoProduction * self myMarket pricePotato)
+ (self myRiceProduction * self myMarket priceRice)
```

Exchanges (Message)

9. askWaterAcquaintances

“select someone among the acquaintances and send a message to request water”

```
| waterRequested m a |
a := peopleToAsk first.
waterRequested := self waterToBeExchanged abs.
m := Exchange new.
m sender: self.
m receiver: a.
m symbol: #waterRequest.
m amount: waterRequested.
self sendMessageAsynchronously: m.
peopleToAsk remove: a
```

10. consultMailBox

“check mailbox for messages and pay money or cash for water requested”

```
self mailBox do:
```

```
[:a |
"self id = 6 ifTrue: [self halt]."
a symbol = #waterGiven ifTrue: [self messageWaterGiven: a].
a symbol = #waterRequest ifTrue: [self messageWaterRequest:
a].
a symbol = #laborRequest ifTrue: [self messageLaborRequest:
a].
a symbol = #moneyRequest ifTrue: [self
messageMoneyRequest: a].
```

```

a symbol = #labor ifTrue: [self messageLabor: a].
a symbol = #money ifTrue: [self messageMoney: a]].
self mailBox: OrderedCollection new

```

11. definePeopleToAsk

“Identify people to ask water from the list of acquaintances only and send message”

```

self waterToBeExchanged < 0
  ifTrue:
    [peopleToAsk := Cormas
     mixt: (self acquaintances select: [:a | (self kinship includes: a)
not])]
  ifFalse: [peopleToAsk := OrderedCollection new]

```

12. exchangeWater

“Ask water to acquaintance if water is needed”

```

self mailBox isEmpty
  ifFalse: [self consultMailBox]
  ifTrue:
    [(self waterToBeExchanged < 0 and: [self peopleToAsk
isEmpty not])
  ifTrue: [self askWaterAcquaintances]]

```

13. messageLabor: a

“Calculate labor for exchange and send message”

```

self laborToBeExchanged: self laborToBeExchanged + a amount

```

14. messageLaborRequest: a

“Receive labor, pay wage and update the income”

```
| m |
```

```

self waterToBeExchanged: self waterToBeExchanged + a amount.
self waterExchanged: self waterExchanged + a amount.
self laborToBeExchanged > 0
  ifTrue:
    [m := Exchange new.
     m sender: self.
     m receiver: a sender.
     m symbol: #labor.

```

```

    m amount: a amount.
self laborToBeExchanged: self laborToBeExchanged - a amount]
  ifFalse:
    [m := Message new.
    m sender: self.
    m receiver: a sender.
    m symbol: #money.
    m amount: 100 * a amount.
    self myIncome: self myIncome - (100 * a amount)].
  self sendMessageAsynchronously: m

```

15. messageMoney: a

“Receive cash and update income”

```
self myIncome: self myIncome + a amount
```

16. messageMoneyRequest: a

“Send message about the cost of each water unit”

```
| m |
```

```

self waterToBeExchanged: self waterToBeExchanged + a amount.
self waterExchanged: self waterExchanged + a amount.
self myIncome: self myIncome - (100 * a amount).
m := Exchange new.
m sender: self.
m receiver: a sender.
m symbol: #money.
m amount: 100 * a amount.
self sendMessageAsynchronously: m

```

17. messageWaterGiven: a

“Update water available for exchange”

```

self waterExchanged: self waterExchanged + a amount.
self waterToBeExchanged: self waterToBeExchanged - a amount

```

18. messageWaterRequest: a

“Message water received and money paid”

```

| m waterGiven |
self waterToBeExchanged > 0
  ifTrue:

```

```

[m := Exchange new.
m sender: self.
m receiver: a sender.
waterGiven := self waterToBeExchanged min: a amount.
self waterExchanged: self waterExchanged - waterGiven.
self waterToBeExchanged: self waterToBeExchanged - waterGiven.
self laborToBeExchanged < 0
  ifTrue:
    [m symbol: #labourRequest.
    m amount: waterGiven]
  ifFalse:
    [m symbol: #moneyRequest.
    m amount: waterGiven].
self sendMessageAsynchronously: m]

```

19. sendInKinship

“Send message to give water to kinship in turns”

```

| receivers waterGiven m |
receivers := self kinship select: [:a | a waterToBeExchanged < 0].
receivers do:
  [:a |
  self waterToBeExchanged > 0
  ifTrue:
    [waterGiven := self waterToBeExchanged min: a waterToBeExchanged abs.
    m := Exchange new.
    m sender: self.
    m receiver: a.
    m symbol: #waterGiven.
    m amount: waterGiven.
    self sendMessageAsynchronously: m]]

```

Appendix 2. Rainfall pattern used in Limbukha model.

Dominantly low rainfall pattern = 60% of the Cycle < 112mm/month
= 40% of the Cycle > 255mm/month

Dominantly normal rainfall pattern = 60% of the Cycle > 255mm/month
= 40% of the Cycle < 112mm/month

1 Time step = 2 Cycles
Cycle 1 = January to Mid-June
Cycle 2 = Mid-June to December

Average rainfall days (1985-2001)

	Months											
	J	F	M	A	M	J	J	A	S	O	N	D
Rainy Days	2	1	5	7	9	15	20	19	15	5	2	1

Average rainfall per day: = 7 mm day⁻¹ (normal rainfall) and
= 3.5 mm day⁻¹ (low rainfall)

Rainfall pattern used in Limbukha model

Time Steps	R1= Dominantly Low		R2 = Dominantly High	
	Cycle 1	Cycle 2	Cycle 1	Cycle 2
1	Low	Low	Normal	Low
2	Low	Normal	Low	Normal
3	Normal	Low	Normal	Low
4	Low	Normal	Low	Normal
5	Low	Normal	Normal	Low
6	Low	Low	Normal	Low
7	Low	Normal	Normal	Normal
8	Normal	Low	Normal	Low
9	Normal	Normal	Normal	Normal
10	Low	Normal	Low	Normal
11	Low	Low	Low	Low
12	Normal	Low	Normal	Low
13	Low	Normal	Normal	Low
14	Low	Low	Normal	Normal
15	Normal	Low	Normal	Low
16	Low	Normal	Normal	Low
17	Low	Low	Low	Low
18	Low	Normal	Low	Normal
19	Normal	Low	Normal	Low
20	Low	Normal	Low	Normal

Appendix 3. Data generated from 36 scenarios of Limbukha Model

Network	Rainfall pattern + Protocol											
	11	12	13	14	15	16	21	22	23	24	25	26
Units of unused irrigation water												
N1	6.7	3	1	6.8	6.8	4.5	7.3	3	1	6.6	7.1	4.05
N2	7.3	7.3	1.1	6.9	3	0	3	3	1.2	3	3	0.6
N3	6.9	6.35	0	7.5	3	0	7.4	3	0.2	3	3	3
Units of water exchanged												
N1	0	0	0	0	0	0	0	0	0	0	0	0
N2	0	0.8	0	0	0	0	0	0.4	0	0	0	0
N3	0	2.2	1.6	23.5	0	0	0	0.4	2.6	0	0	0
No. of plots planted with potato												
N1	6	9	9	6	6	9	11	8	6	5	9	7
N2	11	11	9	7	7	6	6	6	10	9	8	5
N3	7	9	6	8	9	6	11	8	11	10	6	10
No. of plots planted with rice												
N1	59	42	46	59	59	63	59	42	46	59	59	63
N2	59	59	65	59	42	48	42	42	65	42	42	65
N3	59	58	64	53	42	48	59	42	63	42	42	42
Annual income (US\$)												
N1	14.1	6.0	6.5	15.4	15.4	17.4	17.6	6.0	5.8	14.0	16.9	17.0
N2	18.6	17.6	17.7	15.2	5.5	6.1	5.6	5.9	19.3	6.1	6.1	14.8
N3	15.6	16.7	16.0	14.2	6.2	6.2	19.1	5.8	20.6	6.3	5.4	6.3

Notes:

N₁: Only among kinship

N₂: Among all members of same village (first with kinship and then with acquaintances)

N₃: Among members of both the villages (all kinship and acquaintances)

11: Dominantly Low + Exchange water only with kinship

12: Dominantly Low + Exchange water against labor and cash

13: Dominantly Low + Exchange water with kinship and Exchange labor against cash

14: Dominantly Low + Exchange water free of charge

15: Dominantly Low + Exchange labor against water

16: Dominantly Low + P₁ + P₂ + P₃

21: Dominantly High + Exchange water only with kinship

22: Dominantly High + Exchange water against labor and cash

23: Dominantly High + Exchange water with kinship and Exchange labor against cash

24: Dominantly High + Exchange water free of charge

25: Dominantly High + Exchange labor against water

26: Dominantly High + P₁ + P₂ + P₃

Abbreviations

CBNRM	Community-based natural resource management
CIRAD	Centre de coopération internationale en recherche agronomique pour le développement. (Agricultural Research Centre for International Development)
CORMAS	Common-pool Resource and Multi-Agent Systems
CPR	Common Pool Resource
DYT	Dzongkhag Yargey Tshogtshung (<i>District Development Committee</i>)
GYT	Geog Yargey Tshogtshung (<i>Block Development Committee</i>)
ha	Hectare
IDRC	International Development Research Center
IWMI	International Water Management Institute
km	Kilometer
Ls ⁻¹	Liter per second
MAS	Multi-agent system
MoA	Ministry of Agriculture
MoHA	Ministry of Home Affairs
NRM	Natural Resource Management
Nu.	Ngultrum (1 US\$ = Nu. 45.01)
PCS	Planning Commission Secretariat
RGOB	Royal Government of Bhutan
RNRRC	Renewable Natural Resources Research Center
RPG	Role-playing Game
t	Ton
t ha ⁻¹	Ton per hectare
US	United State of America

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

Copyright© by Chiang Mai University

All rights reserved

Glossary

Chatro:	Category of farmer who get half of Cheep's share of water
Cheep:	Category of farmer who get half of Thruelpa's share of water
Chukor:	Rotations of irrigation turns
Langdo:	Unit of land which is equal to 0.1 ha.
Lhangchu:	Category of farmer who do not have access to water.
Mixed Agriculture	Arable land used for growing multiple crops, e.g. kitchen garden where mix of vegetables is grown in small plots.
Rimdo	Annual religious ceremonies performed at household and community level
Shokshing:	Woodlot on which either individual or the community have right-to-use for leaf litter and dry firewood.
Thruelpa:	Originally tax payer in the community. Category of farmer who have full access to water.

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

Copyright© by Chiang Mai University

All rights reserved

CURRICULUM VITAE

Name: Tayan Raj Gurung

Date of birth: March 8, 1961

Educational background:

1982-1986 B.S. Agricultural Science, College of Agriculture,
Kerala Agricultural University, Vellayani, Trivandrum,
Kerala, India.

2002-2004 M.S. Agriculture (Agricultural Systems)
Chiang Mai University, Chiang Mai, Thailand

Scholarships:

Department of Technical and Economic Cooperation
(DTEC), Thailand

Working experiences:

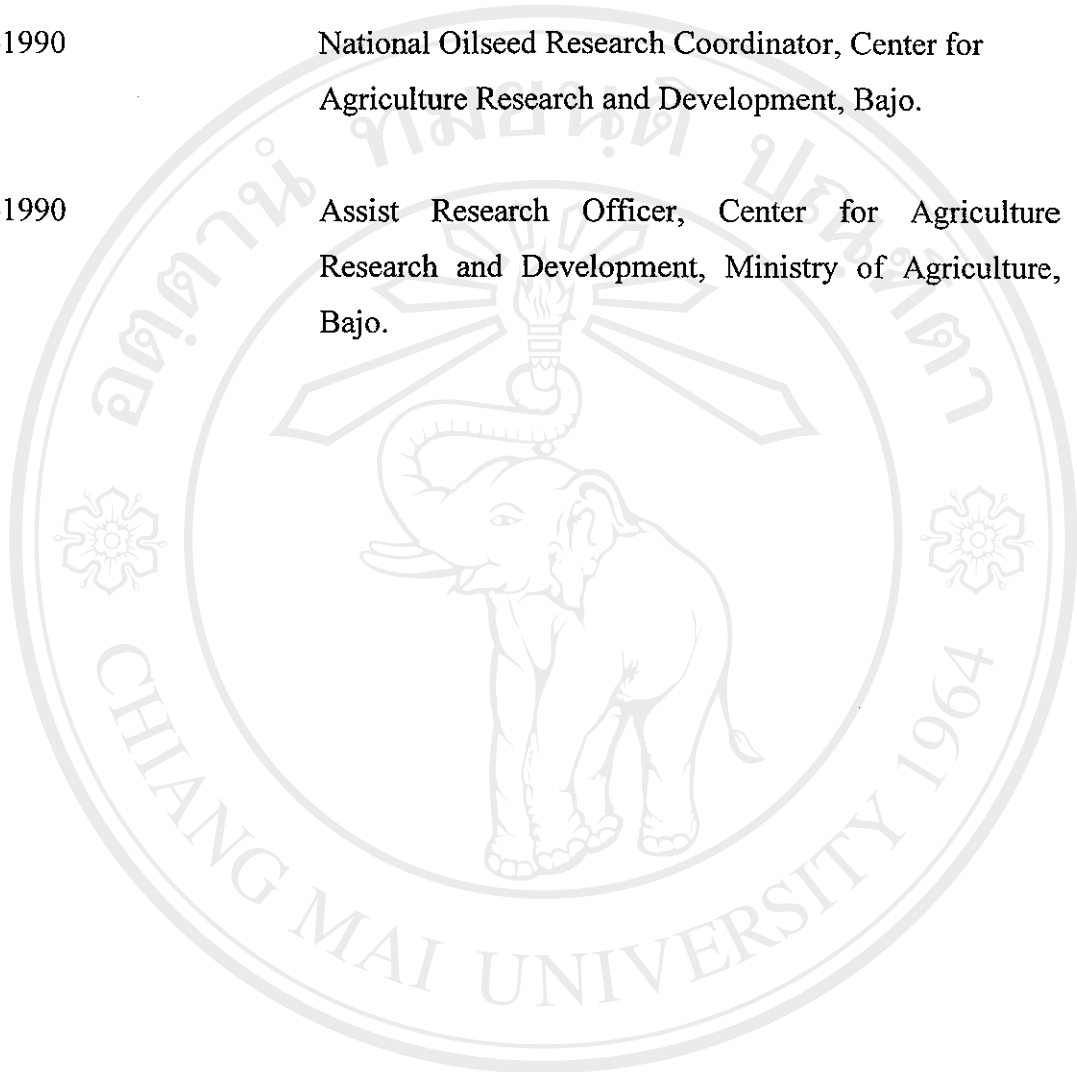
2002-present Program Officer, Local Development Initiatives,
Council for Research and Extension, Ministry of
Agriculture, Thimphu.

1999-2002 Head, Information management section, Council for
Research and Extension, Ministry of Agriculture,
Thimphu

1996-1999 Head, Farming Systems and Support Service Sector,
RNR Research Center, Ministry of Agriculture,
Khangma.

1992-1996 Research Officer, Farming Systems Sector, RNR
Research Center, Ministry of Agriculture, Bajo.

- 1990-1992 Officer In-charge, Center for Agriculture Research and Development, Ministry of Agriculture, Bajo.
- 1989-1990 National Oilseed Research Coordinator, Center for Agriculture Research and Development, Bajo.
- 1987-1990 Assist Research Officer, Center for Agriculture Research and Development, Ministry of Agriculture, Bajo.



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved