QUANTICOL

A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours

http://www.quanticol.eu

TR-QC-03-2017

CARMA for food security: a progress report

Revision: 0.1; March 2017

Author(s): Vashti Galpin (UEDIN), Daniël Reijsbergen (UEDIN)

Publication date: March 2017

Funding Scheme: Small or medium scale focused research project (STREP)
Topic: ICT-2011 9.10: FET-Proactive 'Fundamentals of Collective Adaptive Systems' (FOCAS)
Project number: 600708

Coordinator: Jane Hillston (UEDIN)
e-mail: `Jane.Hillston@ed.ac.uk`
Fax: +44 131 651 1426

Part. no.	Participant organisation name	Acronym	Country
1 (Coord.)	University of Edinburgh	UEDIN	UK
2	Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo"	CNR	Italy
3	Ludwig-Maximilians-Universität München	LMU	Germany
4	Ecole Polytechnique Fédérale de Lausanne	EPFL	Switzerland
5	IMT Lucca	IMT	Italy
6	University of Southampton	SOTON	UK
7	Institut National de Recherche en Informatique et en Automatique	INRIA	France

COOPERATION

Abstract

This document describes the initial progress made in using Carma to model food security at the level of farm production, using two agent-based models from the literature as a starting point. The aim is to assess whether Carma is suitable for expressing these scenarios. These scenarios are based on a single village, and we consider whether the basic models can be extended to a country-wide model, taking into account the possibility of redistribution food between subsistence farmers to address food security. Another aim is to assess whether a methodology can be developed for this type of modelling in Carma.

Contents

1 Introduction

Food security is an important issue globally. The United Nations Committee on World Food Security[1] defines it as follows.

> *Food security is the condition in which all people, at all times, have physical, social and economic access to sufficient safe and nutritious food that meets their dietary needs and food preferences for an active and healthy life.*

Climate variation affects the production of food, and can lead to insufficient food stocks. Tools such as forecasts that enable individuals and organisations to plan for these variations may ensure that sufficient food is produced.

The models described in this report express the behaviour of small-scale farmers in southern Africa, with respect to variations in climate. The original agent-based models were proposed by Ziervogel and her colleagues [ZC03, ZBWD05, BBD+05], and we consider how to create more complex models from these basic models.

The aim of this report is to document our approach to model construction using the modelling language Carma [LH16], a quantitative language based on stochastic process algebra [Hil96, CH09, FH14] and attribute-based communication in process algebra [DLPT14, ADL16, FH14]. Carma has been developed to model collective adaptive systems (CAS) and these farm-level production scenarios can be considered to be examples of this type of system since they consist of a large number of different agents or components trying to achieve goals through interaction and local knowledge in an open-ended environment that requires adaptivity to change.

The structure of the report is as follows: first, we consider the original agent-based models and give an overview of these models. Next we consider the Carma models we have constructed from these models and the validation we have performed to ensure that they provide similar output to the original systems. We then consider one model for expansion. This is the model of a village in the Quthing district in Lesotho, and we expand it to consider production throughout the ten districts of Lesotho. Furthermore we consider the data we can obtain that is district-specific and how this can be used to parameterise a national model of production with grain redistribution. Finally, we consider how using Carma can provide a methodology for this type of modelling.

[1] `http://www.fao.org/cfs/en/`

Figure 1: Turn-off to Mangondi (left) and the main road in Quthing near Ha Tlhaku (right). Images: `maps.google.com`

2 Agent-based models of food production

Two journal article have been published relating specifically to farm/household level of modelling. The first considered market gardeners in Mangondi in the Limpopo province of South Africa [BBD+05] and the second subsistence farmers in Ha Tlhaku in the Quthing district of Lesotho [ZBWD05]. Figure 1 provides pictures of the areas.

Both models are based on fieldwork in the respective villages, and both consider climate variations, accuracy of forecasts and an individual-based trust model of weather forecasts. The South Africa research also considers climate change whereas the Lesotho research does not. Since the South African model considers market gardeners who sell their produce, there is an explicit monetary element to the model. By contrast, the Lesotho model is about subsistence farmers and the cost of insufficient production is measured in term of additional grain needed (rather than the monetary cost of that grain). In both cases, the trust model is based on fieldwork [ZC03].

The basic structure of both scenarios can be described by the following cycle.

1. The weather is determined and a forecast is calculated using accuracy parameters. The weather is described as rainfall using the categories *below normal* (BN), *normal* (N), and *above normal* (AN).
2. The farmers plant crops, taking into account the forecast, their trust in forecast or funds available
3. The farmers harvest their crops, and yield is determined from the actual rainfall.
4. The farmers consume, store or sell their produce.
5. If there is more than one harvest, farmers repeat from step 2.
6. The farmers update their trust in the forecast based on the actual rainfall, the forecast, and other criteria.
7. The year ends and the cycle restarts from step 1.

3 Carma and CaSL

The reader is referred to Deliverables 4.2 and 4.3 of the QUANTICOL project for details of the language. Carma (Collective Adaptive Resource-sharing Markovian Agents) is the formal mathematical language defined for modelling collective adaptive systems [LH16] while CaSL (Carma Specification Language) [LH16, GGG+17] is the language used in the Carma Eclipse Plug-in, the software that embodies Carma and its capabilities [HL16]. The Plug-in allows one to edit models and to perform simulations on them, and it will be used to assess the models developed. Furthermore, it is integrated with MultiVeStA, statistical model checking software which permits probabilistic querying of the trajectories of discrete event simulation [SV13] which we plan to use for more complex queries on the final national model.

The semantics of Carma models are expressed as (time-inhomogeneous) continuous-time Markov chains (CTMCs) [LH16] since Carma is quantitative. CTMCs consider the passing of time through the use of rates which determine the exponential distribution from which action durations are drawn.

CaSL specifications are characterised by the following elements:

Components These comprise a store of attributes to capture the knowledge a component has (and this is typically local knowledge specific to the component); a behaviour, defined in a process algebra style syntax consisting of agents which are states and actions that represent a transition from one state to another state; and an initial state.

Actions Actions are complex and consist of a name with an associated rate; a guard over attribute values which must be true for communication to take place; input or output of values; and an update which can change attribute values in the store. There are four actions prefixes, and in the description below, `e1,e2,..,eN` refer to arbitrary expressions and `x1,x2,..,xN` refer to variables. The presence of `*` indicates broadcast communication which is nonblocking, otherwise the communication is unicast and is blocking, meaning that the sender cannot proceed until interaction with a receiver is completed.

Broadcast output: `action_name*[guard]<e1,e2,..,eN>{my.a1:=new_a1;...;my.aM:=new_aM;}`

Broadcast input: `action_name*[guard](x1,x2,..,xN){my.a1:=new_a1;...;my.aM:=new_aM;}`

Unicast output: `action_name[guard]<e1,e2,..,eN>{my.a1:=new_a1;...;my.aM:=new_aM;}`

Unicast input: `action_name[guard](x1,x2,..,xN){my.a1:=new_a1;...;my.aM:=new_aM;}`

Here, `guard` is defined over store attributes, variables names or store attributes of a potential partner in the communication, thus providing a rich interaction mechanism. For actions which are local to a component, and hence involve no other component, we use

```
action_name*[false]<>{my.a1:=new_a1;...;my.aM:=new_aM;}
```

which describes a broadcast for which no other component can satisfy the guard.

Collective This is the collection of all components involved in a system. There is an initial specification of the collective in a CaSL specification. It is possible for components to leave the collective and disappear, as well as to be added to the collective via an update that is triggered by an action.

Environment The environment of a system consists of a global store together with information about the action rates, probabilities of a broadcast being received, and weights that determine the probability of a specific unicast input matching a unicast output. It also two type of updates that can happen when a specific action occurs: a modification of values in the global store and the addition of new components to the collective.

Measures These define quantities of interest to the modeller, for example, the number of components in a particular state with a specified attribute value. Simulation of CaSL specifications result in trajectories for selected measures.

Functions It is possible to define functions to calculate values of interest, and this how decision making and adaptivity can be captured in a model.

Types CaSL provides three basic types, real, integer and boolean, as well as records and enumerated types. Furthermore, lists and sets can be defined over these types.

Space CaSL provide explicit syntax to describe discrete space. This allows for a parameterised definition of nodes and connections between nodes. Each component has an location attribute if space has been defined.

4 The basic CaSL models

As described earlier, both scenarios have similarities, and this suggest a particular structure for the CaSL description where each household to be modelled is a separate component. An immediate observation from the agent-based models is that action durations are not required and hence we make all actions very fast, so in effect they are happening instantaneously.

Next, it is necessary to have some way to repeat the yearly cycles and there are a number of ways to achieve this. The two most obvious are as follows.

```
component Household (Household_type init_household, real init_stock,
                     int hh_size, int nm_fields)
  store{
    attrib household_type := init_household;
    attrib food_stock := [newer:=RND(init_stock, sqrt(0.2)), old:= 0.0,  older := 0.0];
    attrib household_size := hh_size;
    attrib num_fields := nm_fields;

    attrib trust := 2.0;
    attrib base_rate_of_trust_changes := 1.0;
    attrib disaffection := 0;

    attrib weather := UK;
    attrib forecast := UK;
    attrib field := empty_field;
  }
  behaviour{
    GetRainfallPrediction = rainfall_info*[true](r,f)
        {weather:=r;forecast:=f;}.PlantCrops;
    PlantCrops = plant_fields*[false]<>
        {field:=Plant_fields(num_fields,forecast,trust);}.HarvestCrops;
    HarvestCrops = harvest_crops*[false]<>
        {food_stock := [newer:=Harvest_fields(weather, field),
         old:=food_stock.newer,older:=food_stock.old];}.ConsumeFood;
    ConsumeFood = consume_food*[false]<>
        {yield := food_stock.newer;
         food_stock := Consume_food(food_stock.newer, food_stock.old, food_stock.older,
                       real(household_size * subsist + seed_req));}.UpdateDisaffection;
    UpdateDisaffection = update_sufficiency*[false]<>
        {disaffection := Calc_disaffection(disaffection, food_stock.newer + food_stock.old
                         + food_stock.older <= 0.0, forecast, weather);}.UpdateTrust;
    UpdateTrust = update_trust*[false]<>
        {trust := Calc_trust(trust, disaffection, base_rate_of_trust_changes, forecast,
                  weather);}.WaitForEndOfYear;
    WaitForEndOfYear = end_of_year*[true]().GetRainfallPrediction;
  }
  init{GetRainfallPrediction}
}
```

Figure 2: `Household` component for the basic Lesotho model

1. Specify a synchronising component that receives messages from all `Household` components once they have completed their year and advance to the next year by sending a message to all `Household` components notifying them of this. This has two disadvantages, the first being that if there are many `Household` components, then there will be a large amount of communication which will slow down simulation. Furthermore, in the CARMA Eclipse plug-in, the trajectory of each measure is determined at regular intervals over the time period the simulation runs and details of the trajectories may be lost because the year change will not happen at regular intervals.

2. Specify a synchronising component that waits for a fixed amount of time and then broadcasts a message to all `Household` components that the year has ended. This ensures a regular time pattern to year change but requires ensuring that the action duration are short enough that all actions can be completed in the time available.

It is also possible to combine these approaches to obtain regular year changes with notification from each `Household` component. We have gone with the second choice in the interests of efficiency in simulation.

Figure 2 and 3 show the `Household` component for each of the two scenarios. They are not identical but very similar in structure. There are a number of attributes that are set to initial values, and others that are defined to some empty or null value that will be used as the behaviour of the household unfolds. The behaviour in both cases consists of a fixed cycle of states and actions, with some actions leading to updates of attributes via functions. The functions capture the basic decision-making and

```
component Household (Farmer init_farmer, real init_wealth){
  store{
    attrib farmer := init_farmer;
    attrib wealth := init_wealth;
    attrib weather := rainfall_empty;
    attrib forecast := rainfall_empty;
    attrib wrsi := wrsi_empty;
    attrib plant := veg_empty;
    attrib yield := veg_empty;
    attrib trust := 0;
    }
  behaviour{
    H0   = weather_info*[true](w,r,f)
              {wrsi:=w;weather:=r;forecast:=f;}.HP_1;
    HP_1 = plant_first_crop*[false]<>
              {plant:=Plant_first(wealth,farmer,forecast.s1_r,trust);}.HH_1;
    HH_1 = harvest_first_crop*[false]<>
              {yield := Harvest(plant,wrsi.s1_y);}.HS_1;
    HS_1 = sell_first_crop*[false]<>
              {wealth := max(0.0,Calc_wealth(yield,plant,farmer,weather.s1_r,wealth));
               trust := Calc_trust(trust,weather.s1_r,forecast.s1_r)}.HP_2;
    HP_2 = plant_second_crop*[false]<>
              {plant:=Plant_second(wealth,farmer,forecast.s2_r,trust);}.HH_2;
    HH_2 = harvest_second_crop*[false]<>
              {yield := Harvest(plant,wrsi.s2_y);}.HS_2;
    HS_2 = sell_second_crop*[false]<>
              {wealth := max(0.0,Calc_wealth(yield,plant,farmer,weather.s2_r,wealth));
                 trust  := Calc_trust(trust,weather.s2_r,forecast.s2_r)}.HW;
    HW  = complete_year*[false]<>.HE;
    HE  = end_of_year*[true]().H0;
    }
  init{H0}
}
```

Figure 3: `Household` component for the South African model

other aspects of the model such as yield calculation. In both of the original agent-based models, there is no communication between households. Although the South African scenario includes selling and purchase of stocks, this is done through an abstract market mechanism that does not require interaction between households. Likewise, the basic scenario for the Lesotho model has no interaction between households[2].

5 Model validation

Since our CaSL models have been developed from a textual description of previous agent-based models, it is possible that we have misunderstood the details, and validation could be useful to discover this, if possible. For the Lesotho scenario, we have worked from the journal article and reproduced some of the results. This is the approach we also tried for the South African scenario, but it was not possible to ascertain some of the details. We have obtained the Java code for the agent-based model and further work is required to understand this code to finalise the details for which we were uncertain.

In the original paper about the Lesotho smallholders, Figure 4(b) shows the cumulative cost of poor years for a single simulation. This graph is presented in the left of Figure 4. The right of Figure 4 shows the output from our CaSL specification via the CARMA Eclipse Plug-in for the averages over 10 simulations. Inspection suggests that the results are similar but comparison of single simulations is not a strong basis for validation. For both scenarios, there is more validation to be completed.

[2] A more complex model mentioned in the same paper has interaction in terms of a social trust model but we do not consider this here.

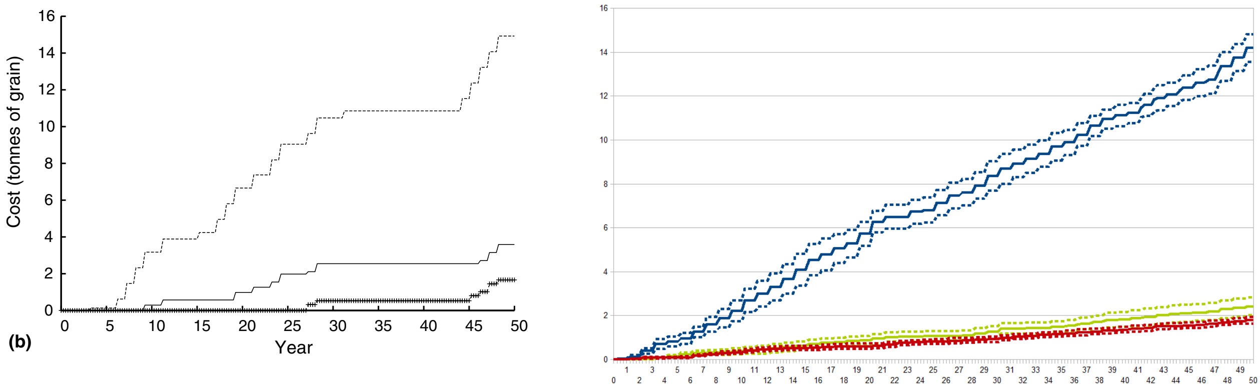

Figure 4: Left: Cumulative cost of poor years. Extra costs incurred as a result of crop failure in three cases. Solid line: no forecast, dotted line: forecasts always incorrect, plus line: 65% correct forecast [ZBWD05]. Right: Cumulative cost of poor years from CaSL specification. Averages over 10 simulations. Green line: no forecast, blue line: forecasts always incorrect, red line: 65% correct forecast. Dotted lines indicate confidence intervals.

6 Development of a national model

We are interested in considering more than a single village, and hence are in the process of investigating whether the construction of a model of the whole of Lesotho would be viable. The aim of such a model would be to investigate whether through redistribution (by government or non-governmental organisations) or a market mechanism, there is sufficient grain grown to ensure that households do not incur extra costs due to poor weather circumstances. This involves a number of steps.

1. Create a relatively naive national model in CaSL and assess what sort of variation in production there is across the ten districts of Lesotho, and whether it seems likely that redistribution (through whatever mechanism) could increase food security.

2. Add redistribution to the model, and assess results. There are two ways in which this can be done: ignoring space and distance, and assuming that there is no cost to redistribution of food stocks regardless of where they are in the country; and taking space and distance into account and ensuring that redistribution only happens between neighbouring districts. We have implemented the space-homogeneous scenario as a first step.

3. Collect district-based parameters and aim for a more realistic model.

The next section discusses the model extensions and the preliminary results that were obtained. The section after that considers data sources for parameters for the last step.

7 Preliminary results: step 1

The first step was achieved by creating a space specification in CaSL where each district is a distinct node, and connections describe the road links between them (which is not simply adjacency of districts due to the mountainous terrain in Lesotho and small road network). A fixed household size was used and each `Household` components was allocated to a district, in proportion with the population of each district from the 2006 Census[3]. Additionally weather data was obtained for each district, resulting in 41 sets of data for years in the period between 1993 and 1977 from KNMI Climate Explorer[4] and one

[3] `www.bos.gov.ls/2006_census.htm`

[4] Koninklijk Nederlands Meteorologisch Instituut, `climexp.knmi.nl`

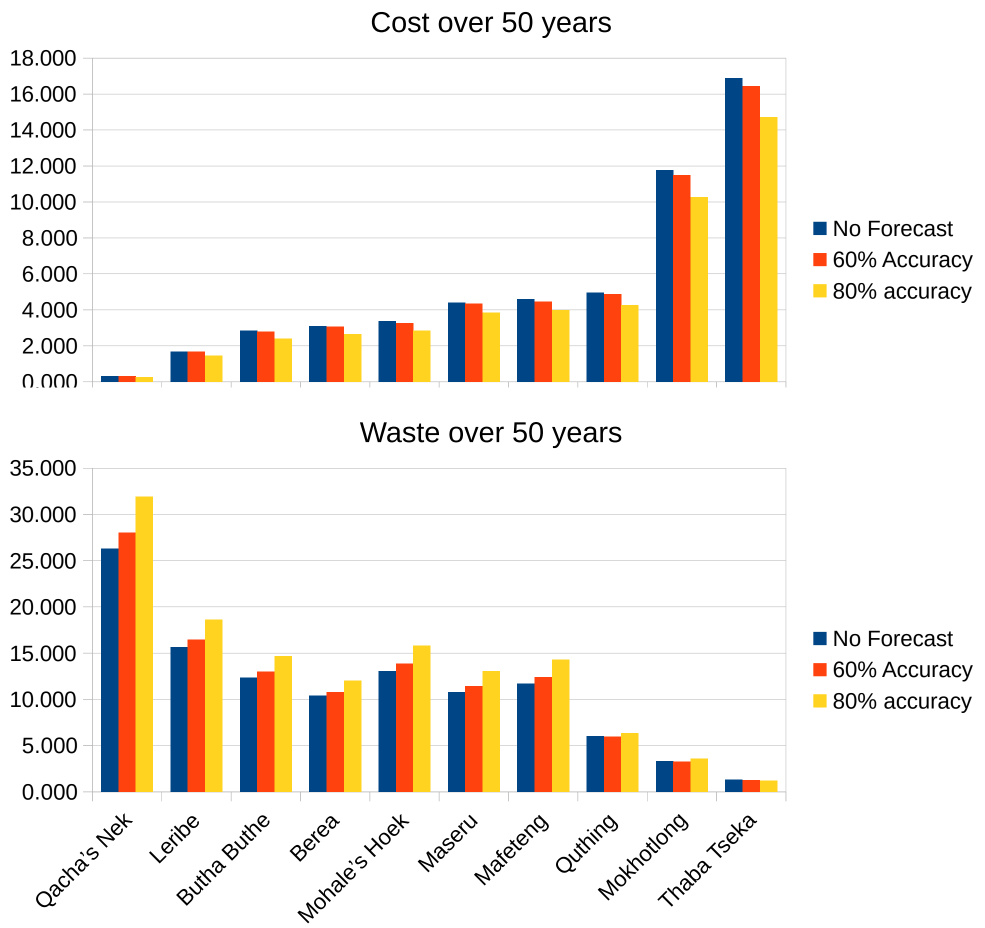

Figure 5: Cumulative cost and waste for each district in Lesotho over a fifty year period for different forecast accuracies.

of these sets is selected at random for each year cycle. Figure 5 shows the results from this model. The upper graph shows the cost (in terms of extra food required) for each of the ten districts for no use of forecast and two different forecast accuracies. It can be seen that there is a wide variation between districts, and that use of more accurate forecasts leads to lower cost. The lower graph shows the waste in each district. Waste occurs when stored grain that has not been consumed must be discarded after two years of storage. Again, it can be seen that here is wide variation in waste, and waste increases with forecast accuracy. These results suggest that redistribution of grain between districts could be a viable strategy for food security.

8 Preliminary results: step 2

The second step involves redistribution and requires additional components to be added to the model. These add a centralised aspect of the model with components having knowledge that they collect from `Household` components and then use to determine redistribution and to feed it back to `Household` components. `Household` components must be modified to communicate their food stocks and receive results after the communication.

A `District` component is added for each district together with a single `National` component. Their `CaSL` specifications are given in Figure 6. A `District` component gathers information from each household in that district. Here it is assumed that each household in a district is allocated a consecutive number from 1 to the number of households in that district but this is not the only approach to collecting

```
component District (int n){
  store{
    attrib num = n;
    attrib fi = [:empty_food_info:];
    attrib total_f = empty_food_info;
    attrib i := 1;
  }
  behaviour{
    Rec = [i<=num]send_info[my.loc==l&&my.i==id](l,id,f)
              {fi:=fi+[:f:];total_f:=plus(total_f,f);i:=i+1}.Rec +
          [i==num+1]combine_nat[true]<my.loc,total_f>
              {i:=1;}.Get;
    Get  = split_nat[my.loc==[l]](l,sf)
              {fi:=Realloc_food(fi,sf,total_f,num);i:=1;}.Send;
    Send = [i<=num]send_food[true]<my.loc,i,fi[i]>
               {i:=i+1;}.Send +
           [i==num+1]send_complete*[false]<>
               {fi:=[:empty_food_info:];total_f:=empty_food_info;i:=1}.Rec;
  }
  init{Rec}
}

component National (){
  store{
    attrib num = num_districts;
    attrib nfi = [:empty_food_info:];
    attrib total_nf = empty_food_info;
    attrib j := 1;
  }
  behaviour{
    RecN = [j<=num]combine_nat[l==[my.j]](l,rf)
              {nfi:=nfi+[:rf:];total_nf:=plus(total_nf,rf);j:=j+1;}.RecN +
           [j==num+1]reallocate*[false]<>
              {nfi:=Realloc_nat(nfi,total_nf,num);j:=1}.SendN;
    SendN = [j<=num]split_nat[true]<j,nfi[j]>
               {j:=j+1;}.SendN +
            [j==num+1]send_nat_complete*[false]<>
               {nfi:=[:empty_food_info:];total_nf:=empty_food_info;j:=1}.RecN;
  }
  init{RecN}
}
```

Figure 6: `District` and `National` components

this data. Guards in actions are used to ensure that the correct data is collected. Then each district communicates its total to the `National` component in a similar fashion. The function `Realloc_nat` is called to reallocate stocks between districts and the reallocation communicated to districts, and then to households.

There are obviously a number of reallocation functions that can be used. The one we have implemented is a "no cost" function since it reallocates any two year-old grain that cannot be consumed and would be wasted in the next year. Figure 7 illustrates how the reallocation of grain between districts affects the number of years in which there is a food shortage, for two different trust models and two different forecast accuracies. The two trust models are the original model from [ZBWD05] and a modified model that decreases the likelihood of taking the forecast into account. The graph illustrates that the number of years with unsatisfied demand for grain can be reduced through distribution, although not to zero, using the reallocation of unused grain that will go to waste. Note that this model is space-homogeneous in that assumes that transfer between districts is equal for all pairs of districts. This result gives support for the next step of identifying parameters to produce a more realistic model.

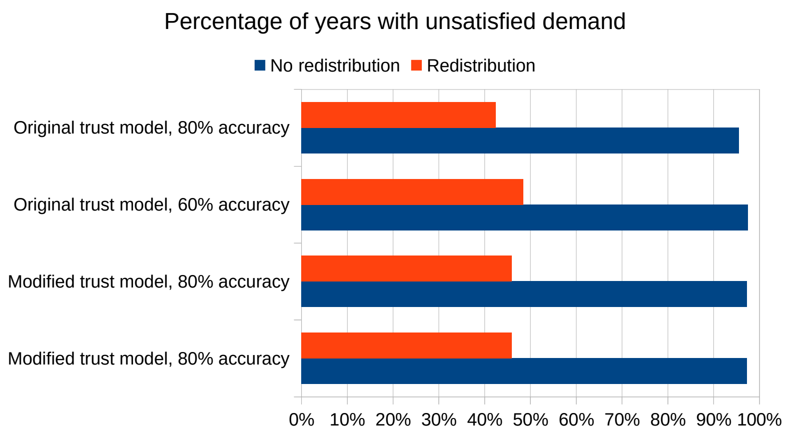

Figure 7: Percentage of years with unsatisfied demand: average over 1000 simulation at year 15

Parameter/data	Source
Planting decisions	Local fieldwork
Trust model	Local fieldwork
Crop yields	National data
Rainfall data	National data
Household sizes	National data
Number of fields	National data
Field size	Unclear
Area planted per district	Not required

Table 1: Model parameters and source for original agent-based model.

9 Parameters and data: step 3

The original paper [ZBWD05] for the Lesotho model is based on a single village in the district of Quthing, in the south-east of the country. Some of the data such as the yield data are nationally-based, whereas the trust model was developed locally to the specific village in which the fieldwork took place [Zie04]. In extending the whole model to Lesotho, it would ideal to obtain figures and parameters for each districts. Table 1 identifies that data that is needed together with what was used in the original agent-based model. Except for the fieldwork, most data is national. It would be possible to use this data in each district but it would be better to use district-level data if it is available, as we are aiming for veracity. The field size is assumed to 1ha and it is not clear from where this was obtained. The final item was not required because the model was district-specific rather than including all districts. We now describes potential data sources.

9.1 Planting decisions and trust models

The details of these were obtained during fieldwork in Lesotho, and it is very improbable that similar fieldwork has been carried out in other districts. Hence, this is best data available and will be used. In Quthing district, the grains planted are maize and sorghum but in other districts, wheat is planted rather than sorghum, so either what should be added as a crop, or there should be two crops: maize and "other crop". Even in the case when wheat and sorghum are grouped together it may be useful to use different yield profiles.

9.2 Area planted to crops per district

The Lesotho Bureau of Statistics[5] publishes various reports including an annual survey of agricultural crop production for which a number of years are easily accessible online [Les14, Les13, Les11, Les09]. These detail how much land is planted for each crop type in each district. By taking data across a number of years, an average can be obtained for each district. It may also be possible to determine trends from this data and hence predict future plantings.

9.3 Household sizes, number of fields and field size per district

These items are not covered in the annual survey of agricultural production, but a 10-yearly agricultural census is performed that covers this data with the currently available one from 2009/10 [Les12].

9.4 Crop yields per district

The annual survey of agricultural production for crops includes yield data for each district. However, what is required is some way to take rainfall data and obtain yield data from it. This type of equation is available nationally and was used in the agent-based model but it may be possible to develop equations specific to each district if one can combine the yield for each district for a number of years together with information about rainfall.

9.5 Annual rainfall per district

As described above, it has been possible to find data for 41 years up to 1997 of rainfall in each district, However, as noted in the previous point, this ideally needs to be correlated with yield data, but the available yield data is more recent. Although there is a fair amount of historic weather data available for Lesotho, this particular data is not available. Various unsuccessful attempt have been made to obtain data, including contacting the Lesotho Meteorological Services[6]. Currently the best source appears to be graphs in a report on climate change and health [Che12].

This is a preliminary assessment of data sources and the next steps are to seek out more data if possible, then to extract the data from the sources, which in some cases may involve transformation of the data; and finally, development of appropriate data types and functions in the CaSL specification. Furthermore, the correlation of rainfall data and yield by district should be done if the appropriate data can be found.

10 Conclusions

This report has documented the progress that has been made in developing a national farm-level model of food security for Lesotho, as well as highlighting the work still to be done to obtain a parameterised model. This research raises the question of whether this work leads to a proposal for a methodology for modelling food security at this level in Carma and CaSL. Figure 8 suggests a basic generic household component that could be used in such a methodology. This is a partial specification because it does contain all the variables that might be necessary such as household size, but it is not obvious how to ensure that all the necessary variables are included. The behaviour includes a loop for multiple harvests, with the index `h` being incremented and passed as an argument to the various harvest-dependent functions. However, assumptions have been hard-coded into the specification, such as trust being updated after every harvest and food redistribution taking place only once a year. This is not ideal, and a better approach may be to provide a domain specific language for a food security modeller to use that would provide the necessary flexibility.

[5] `http://www.bos.gov.ls`
[6] `http://www.lesmet.org.ls`

```
component Household (real init_trust,real init_wealth,...){
  store{
    attrib trust := init_trust;
    attrib wealth := init_wealth;
    attrib ... := ...;
    attrib weather := rainfall_empty;
    attrib forecast := rainfall_empty;
    attrib plant := veg_empty;
    attrib yield := veg_empty;
    attrib stocks := stock_empty;
    attrib h := 0;
    }
  behaviour{
    Start   = weather_info*[true](r,f)
                 {weather:=r;forecast:=f;}.Plant;
    Plant   = plant_crop*[false]<>
                 {plant:=Plant(h,forecast,trust,wealth...);}.Harvest;
    Harvest = harvest_crop*[false]<>
                 {yield := Harvest(h,plant,weather,...);}.Consume;
    Consume = consume_store*[false]<>
                 {stocks:=ConsumeStore(h,stocks,...);}.Sell;
    Sell    = sell_food*[false]<>
                 {wealth:=SellBuy(h,stocks,wealth);}.Trust;
    Trust   = update_trust*[false]<>
                 {trust:= Calc_trust(trust,weather,forecast,stocks,...)}.Loop;
    Loop    = [h<num_harvests]repeat*[false]<>{h:=h+1;}.Plant +
              [h=num_harvests]continue*[false]<>{h:=0;}.Send;
    Send    = send_info*[true]<id,stocks>.Rec;
    Rec     = send_food*[my.loc==l&&my.id==i]<l,i,s>
                  {stocks:=UpdateStocks(s,stocks).Start
    Finish  = end_of_year*[true]().Start;
    }
  init{Start}
}
```

Figure 8: Generic `Household` component

References

[ADL16] Y.A. Alrahman, R. De Nicola, and M. Loreti. On the power of attribute-based communication. In *Proceedings of the 36th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE 2016)*, LNCS 9688, pages 1–18. Springer, 2016.

[BBD+05] S. Bharwani, M. Bithell, T.E. Downing, M. New, R. Washington, and G. Ziervogel. Multi-agent modelling of climate outlooks and food security on a community garden scheme in Limpopo, South Africa. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360:2183–2194, 2005.

[CH09] F. Ciocchetta and J. Hillston. Bio-PEPA: a framework for the modelling and analysis of biological systems. *Theoretical Computer Science,*, 410:3065–3084, 2009.

[Che12] R.A. Cheke. Report on a consultancy for the Africa Adaptation Programme on possible climate change effects on health in Lesotho, 2012. `http://www.undp-aap.org/resources/projects/aap-lesotho-technical-report-climate-change-effects-health-2012`.

[DLPT14] R. De Nicola, M. Loreti, R. Pugliese, and F. Tiezzi. A formal approach to autonomic systems programming: The SCEL language. *ACM Transactions on on Autonomous and Adaptive Systems*, 9:7:1–7:29, 2014.

[FH14] C. Feng and J. Hillston. PALOMA: A process algebra for located Markovian agents. In *Proceedings of QEST 2014*, LNCS 8657, pages 265–280. Springer, 2014.

[GGG+17] V. Galpin, A. Georgoulas, S. Gilmore, J. Hillston, D. Latella, M. Loreti, M. Massink, and N. Zoń. CASL at work. *QUANTICOL Deliverable D4.3*, 2017. `http://blog.inf.ed.ac.uk/quanticol/deliverables`.

[Hil96] J. Hillston. *A compositional approach to performance modelling.* CUP, 1996.

[HL16] J. Hillston and M. Loreti. CARMA eclipse plug-in: A tool supporting design and analysis of collective adaptive systems. In *Proceedings of the 13th International Conference on Quantitative Evaluation of Systems (QEST 2016)*, LNCS 9826, pages 167–171. Springer, 2016.

[Les09] Lesotho Bureau of Statistics. 2008/09 Agricultural Production Survey - Crops, 2009. `http://www.bos.gov.ls`.

[Les11] Lesotho Bureau of Statistics. 2010/11 Agricultural Production Survey - Crops, 2011. `http://www.bos.gov.ls`.

[Les12] Lesotho Bureau of Statistics. Lesotho Agricultural Census 2009/10, Vol I: Crops, 2012. `http://www.bos.gov.ls`.

[Les13] Lesotho Bureau of Statistics. 2012/13 Agricultural Production Survey - Crops, 2013. `http://www.bos.gov.ls`.

[Les14] Lesotho Bureau of Statistics. 2013/14 Agricultural Production Survey - Crops, 2014. `http://www.bos.gov.ls`.

[LH16] M. Loreti and J. Hillston. Modelling and analysis of collective adaptive systems with CARMA and its tools. In *Formal Methods for the Quantitative Evaluation of Collective Adaptive Systems: 16th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM 2016)*, LNCS 9700, pages 83–119. Springer, 2016.

[SV13] S. Sebastio and A. Vandin. MultiVeStA: Statistical model checking for discrete event simulators. In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '13)*, pages 310–315. ICST/ACM, 2013.

[ZBWD05] G. Ziervogel, M. Bithell, R. Washington, and T. Downing. Agent-based social simulation: a method for assessing the impact of seasonal climate forecast applications among smallholder farmers. *Agricultural Systems*, 83:1–6, 2005.

[ZC03] G. Ziervogel and R. Calder. Climate variability and rural livelihoods: assessing the impact of seasonal climate forecasts in Lesotho. *Area*, 35:403–417, 2003.

[Zie04] G. Ziervogel. Targeting seasonal climate forecasts for integration into household level decisions: the case of smallholder farmers in lesotho. *The Geographical Journal*, 170:6–21, 2004.