Fuzzy multiagent distributed assembly chart planning in agriculture*

Tikhonov E.
Russia, Msk. reg., Dmitrov,
Vnukovsky mkr., 13 45
yevgesh@mail.ru

Abstract

Agriculture in Russia has some specifics such as large distribution, inclement climate and big market competition. Appearing of resources-economy, precision and management technologies demanded from agriculture companies new planning and control methods. The paper is dedicated to assembly charts planning in agriculture with multi-agent approach. For resources distribution we used method of virtual auction and fuzzy logic controller. JACK Intelligent Agents platform is used for implementation.

Keywords: multi-agent technology, production planning, agriculture, fuzzy logic, JACK Intelligent Agents.

1 Introduction

For agriculture production in inclement and unstable climate conditions it’s very important to react intelligent on different unexpected situations. As usual season plans of cultivating crops differs from results at the end of the year. Plans can change some times a day because of climate changes, breakdowns, supply delays or just a human factor.

The main agriculture instrument for planning is assembly chart which helps economists find out expenses and profits of nurturing specific crop. This chart includes land, aggregates, machines, tractors, personal and technological operations. Working the assembly chart up is serious problem especially in large agro-holdings and when taking into account quality figures of aggregates, time component, land condition and changing environment. These requirements make a further stress to the agriculture production planning system, which must be dynamically adaptable to both local and distributed utilization of production resources and materials.

In this work the multi-agent approach applied for agriculture assembly chart planning discussed in [1-3]. Planning system is implementing with Java based JACK Intelligent Agents platform.

2 Model

Agriculture production to stay competitive faces the problem of prime cost reduction using various information about soil, machines, fertilizers. The effects of these trends can be summarized as increasing complexity and the need to respond to continual change under decreasing costs. To meet these new business challenges, manufacturing operations require additional functionality, like robustness, scalability or reconfigurability, while maintaining simple and transparent processes [3].

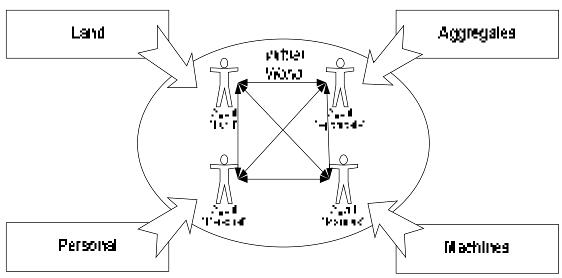

Figure. 1. Presentation entities of model as communicating agents.

Such models could be applied as self-organized intelligent agents in virtual worlds [4] where each agent is an entity of the model with its own properties. And it could cooperate with other agents to achieve own goals.

Let's represent main objects of our assembly chart model with intelligent agents which are able to communicate with each other. For assembly chart main objects are: lands, aggregates, machines and personal (Fig. 1). There could be unlimited number of agents of these types in the model. Main properties of land, aggregate and machine agents shown in figure 2.

Each operation takes special aggregates, machines and personal in order to grow the crop. But the quality and cost of carrying the task out is different on various lands. And the task handling may be done with different quality and cost as well. So we need to plan choosing aggregates and machines for land processing in the way to maximize quality and minimize prime-costs.

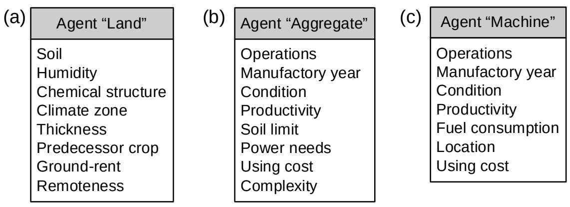

Figure 2. Model objects properties: (a) Land, (b) Aggregate, (b) Machine

This approach can help easily renegotiate, re-plan and make right and coordinated decision.

We included "Expert" and different crops (wheat, potato, grass and etc.) agents as well representing experts in the system.

3 Agent architecture

Among the main requirements of an autonomous agent is the ability to perform means-end reasoning, i.e. the ability to select a course of actions that ultimately achieves the goals of the agent. There has long been a strive in the Artificial Intelligence community for an agent architecture, that enables selection of a course of actions that ultimately achieves the goals of the agent. This has led to many approaches to practical reasoning, where the most notorious and respected of which is the agent model known as Beliefs, Desires and Intentions [5].

The BDI kernel, seen in figure 3, is the pivot and functions as the interpreter. Its execution cycle is as follows: At time t: certain goals are established and certain beliefs are held. An event occurs that alters the beliefs or modify the goals, and the new combination of goals and beliefs trigger plans. If the required capabilities exist, the goal that was triggered by the event, is placed in the intentions structure.

As the main objective of agent is planning we use fuzzy module to determine if the model object represented by agent capable to do necessary tasks based on current conditions. For this purpose the fuzzy logic controller is used.

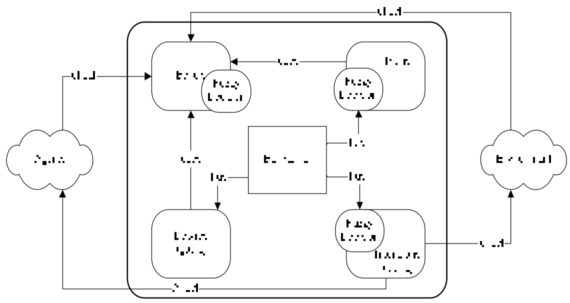

Figure 3. The BDI model with the addition of making fuzzy decision support, interacting agents and the environment.

An executable plan is found and executed, whereby the agent performs the action. Performing the action may change the environment and establish new goals and beliefs, and so the BDI kernel's execution cycle starts again.

3.1 Fuzzy controller

The main purpose of fuzzy logic controller use is to increase flexibility of choosing contractor for agriculture operations maintaining. It is also used to enlarge agent beliefset utilization quality as shown in fig. 3 and fig. 4.

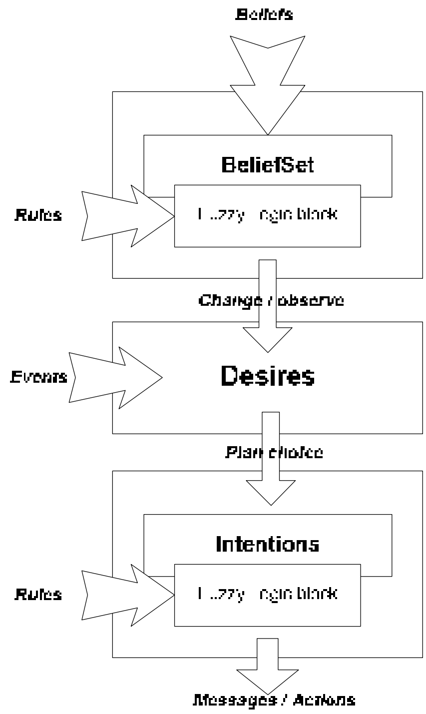

Figure 4. The BDI model with fuzzy logic blocks, introspection steps.

Taking into account the fact that each agricultural technologic operation needs fixed soil condition it is very convenient to use fuzzy logic block with the support of passing needed knowledge as rules. The same situation we observe on aggregates and machines which have common properties but show them in different ways.

Why we use fuzzy logic controller? The answer to this question comes from the fuzziness paradigm where one or more properties have fuzzy limits. This problem corresponds with agriculture. There is also a lot of indeterminism in agriculture mostly because of natural production conditions. Also fuzzy logic tool is very convenient as it allows reducing (some kind of data fusion) to one grade diversified factors (qualitative and quantitative) what is very valuable around agriculture experts (agronomists).

The main principles of fuzzy logic is perfectly described in [6],[7].

Figure 5 shows structure of fuzzy control module used in our agents. It consists of rule base, fuzzification part, inference block, defuzzification part. But using communicating agents allow us on-line download or tune parameters of fuzzy controller. That also allows to take opportunity of available knowledge to improve processes and real-time decision making.

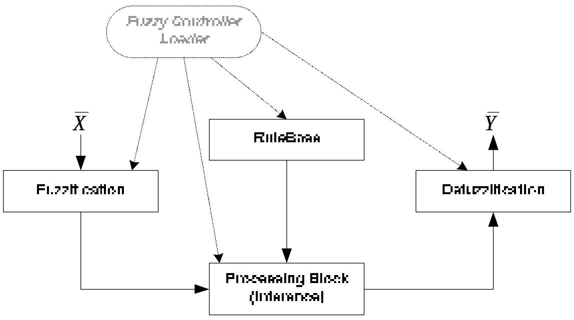

Figure 5. Fuzzy controller.

Fuzzy control engine is build into agent's architecture but setting of system parameters mainly happens during work process depending on current internal state, beliefs and environment condition.

For example, agent "Field *i*" starting agricultural production process with current soil parameters (predecessor, chemical structure, humidity and etc.). It requires some knowledge to decide if soil parameters favorable for starting technological process operations for specific crop. Our agent system divide this problem into five steps as shown in figure 6.

Main steps are:

1. Sending request to expert agent ("Expert") ,
2. Agent "Expert" requesting all known crop agents (wheat, potato, etc),
3. Crop agent estimates suit coefficient based on information attached to request message. This coefficient also may be evaluated using fuzzy logic controller. It is supposed that crop agent has appropriate knowledge about suitable conditions,
4. Sending evaluated coefficients to "Field *i*" agent with necessary knowledge about operation starting parameters for soil,
5. Receiving estimated coefficients and choosing the best one (highest possible). Loading knowledge into fuzzy controller about "crop' and beliefset monitoring for operation starting.

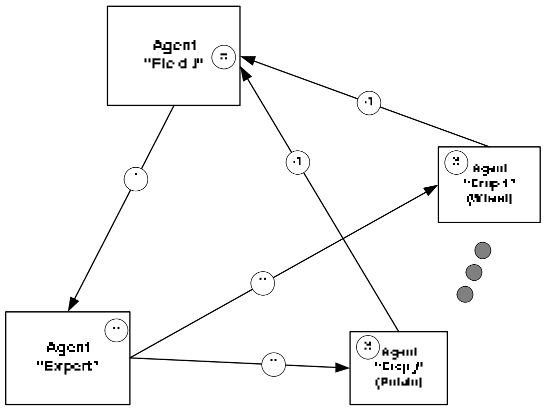

Figure 6. Agent "Field *i*" knowledge obtaining process.

Knowledge passes through agent's network as a set of linguistic terms and variables, membership functions and collection of linguistic rules to attain certain objectives in the form of IF-THEN rule with condition and conclusion.

We used *jFuzzyLogic* package for embedding fuzzy control block into agent. *jFuzzyLogic* is a fuzzy logic package written in java [8]. *jFuzzyLogic* is a java implementation of a Fuzzy Logic software package. It implements a complete Fuzzy inference system (FIS) as well as Fuzzy Control Logic compliance (FCL) according to IEC 1131 [9]. It will allow for real-time packet analysis and can be integrated into other Web-based or Agent-based Network Tools.

The Fuzzy Control applications programmed in Fuzzy Control Language FCL are encapsulated in Function Blocks (or Programs). The Function Block Types defined in Fuzzy Control Language (FCL) shall specify the input and output parameters and the Fuzzy Control specific rules and declarations. The corresponding Functions Block Instances shall contain the specific data of the Fuzzy Control applications.

All descriptions in FCL are enclosed between FUNCTION_BLOCK, END_FUNCTION_BLOCK statements. For example, fuzzification of "humidity" parameter with FCL may be like

```
VAR_INPUT                // Define input variables
    humidity : REAL;
    ...
END_VAR
FUZZIFY humidity         // Fuzzify input variable 'humidity': {'dry' , 'average' , 'damp'}
    TERM dry := (0, 1) (40, 0) ;
    TERM average := (30, 0) (50,1) (60,1) (80,0);
    TERM damp := (70, 0) (90, 1) (100, 1);
END_FUZZIFY.
```

Defuzzification part in FCL also has its method and looks like

```
DEFUZZIFY tip            // Defzzzify output variable 'tip' : {'poor', 'middling', 'best' }
    TERM poor := (0,0) (20,1) (40,0);
    TERM middling := (30,0) (55,1) (80,0);
    TERM best := (70,0) (90,1) (100,0);
    ACCU : MAX;          // Use 'max' accumulation method
    METHOD : COG;        // Use 'Center Of Gravity' defuzzification method
    DEFAULT := 0;        // Default value is 0 (if no rule activates defuzzifier)
END_DEFUZZIFY.
```

Production rules contain into section "RULEBLOCK" and include directions for activation and accumulation.

```
RULEBLOCK No1
    AND : MIN;           // Use 'min' for 'and' (also implicit use 'max' for 'or' to fulfill DeMorgan's Law)
    ACT : MIN;           // Use 'min' activation method
    RULE 1 : IF humidity IS dry AND thickness IS solid THEN tip IS poor;
    RULE 2 : IF humidity IS dry AND thickness IS average THEN tip IS poor;
    RULE 3 : IF humidity IS dry AND thickness IS soft THEN tip IS middling;
    ...
END_RULEBLOCK
```

Actually we can implement much more features as it provided in jFuzzyLogic package: weighting factor, subconditions, various membership functions, defuzzification, accumulation and aggregation methods and etc.

4 Planning process

The plan is created collectively by a community of simple planning agents that use a sophisticated auction-based negotiation, supported by use of the social knowledge and acquaintance models. The core of our system is a community of planning agents which making production plans for individual orders, taking care of conflicts and managing re-planning and plan reconfiguration.

A stable and industry accepted approach to the coordination of agents' joint activity is based on clear cut roles (even temporary) in the multi-agent community. Let us have a coordinator ("field" agent) who is in charge of proper task decomposition and subcontracting contractors for implementing components of the tasks. A classical and industry accepted negotiation algorithm is contract-net-protocol. There is used the simplified version of contract-net in our system.

Any agent (will become a coordinator) can initiate the contract net by requesting some contractors for specific services. Each contractor carries out its own internal reasoning and suggests a collaboration proposal.

Planning process starts at the problem of choosing appropriate crop and sort of the crop according soil conditions on specific lands. It's proposed that land agent knows its main parameters. They can be achieved from experts (agronomists), extracted from previous land use or obtained from special sensors [10]. But land agent can't initiate production process in this situation as it doesn't know anything about what are to grow. So the land agent has to resort to the help of other agents through the communication process.

At figure 6 it's shown how knowledge can be obtained as a set of production rules, linguistic terms and methods for use with fuzzy controller. Fuzzy support in this case is very valuable and helps us to make a reasonable decision taking into account economical, agricultural and ecological factors.

When essential knowledge has loaded agent can proceed to soil condition controlling. This operation lean on data gathered from sensors or received with messages (fig. 7). When auspicious conditions take place event is send and agent starts aggregates selection. Conditions are considered favorable when estimated with fuzzy controller grade after defuzzification reaches given threshold or gets into interval (fig. 8,9).

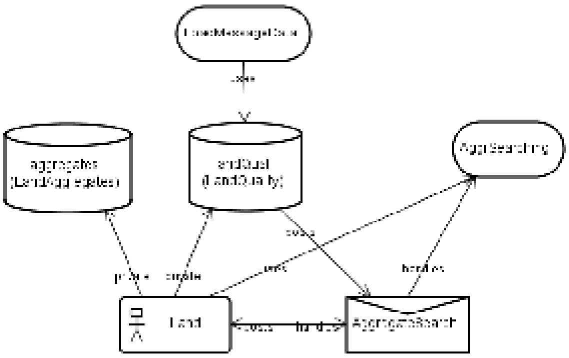

Figure 7. Soil controlling and operations initiation diagram.

Figures 8 and 9 demonstrate our approach for the case when two parameters are controlled (“humidity” and “thickness”) and threshold estimates (“tip”).

Input values are normalized to [1..100] interval and generated randomly. There is defuzzifier chart for the case of values 65 (“humidity”) and 75 (“thickness”) in figure 9.

Classically operation of crop and crops’ sort choosing is done by experts before. As well all operation periods are selected by agronomists.

After choosing the sort and, of the crop and coming appropriate conditions agent starts looking for resources necessary for undertaking the operation. In this case agent initiate choosing plan (“AggrSearching”) through posting event (“AggregateSearch”). The plan is responsible for negotiating all known aggregates (“aggregates”) by sending bids for carrying current operation out (collaboration).

Received message initiates estimate computation based on its own beliefs, obtained operation data and appropriate knowledge (fig. 10) with the help of fuzzy logic controller. With this estimate we can measure the extent of convenience using current aggregate.

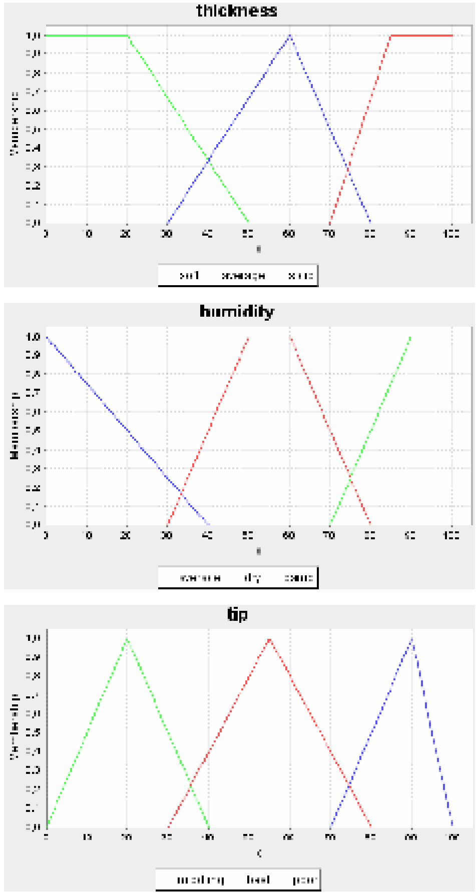

Figure 8. Charts for linguistic variables in the ruleSet: (top) input var. “thickness”, (middle) input var. “humidity”, (bottom) output var. “tip”.

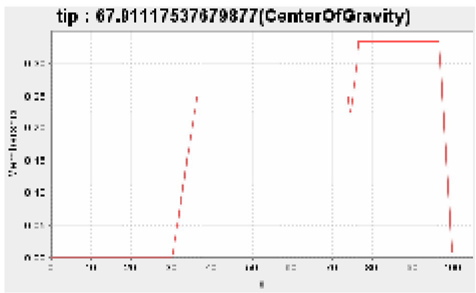

Figure 9. Defuzzifier chart.

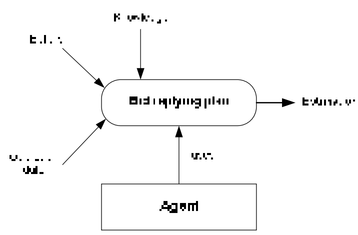

Figure 10. Estimation process diagram.

In picture 11 estimating plan is called “Choose” and after computation it becomes the contactor for sub collaborator finding. It is necessary for the best tractor selection in addition to aggregate (fig. 12).

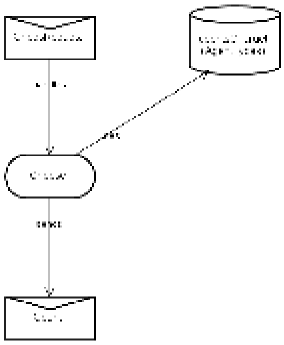

Figure 11. “Data-event-plan” diagram for aggregate agent.

While standard resource distribution approach is based on market mechanism, our estimations point of view allows take qualitative characteristics into consideration.

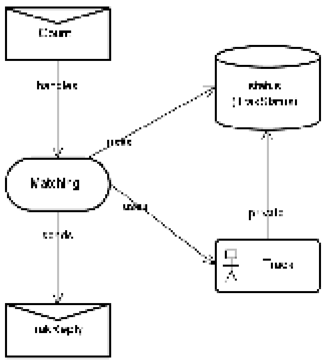

Figure 12. “Data-event-plan” diagram for tractor agent.

Upon receiving proposals for collaboration, the coordinator carries out a computational process by which it selects the best possible collaborator(s) – see Figure 13. The contract net protocol can be also multi-staged. For each single-staged communication within a community of *n* field agents, it is needed to send 2(*n* + 1) messages in the worst case.

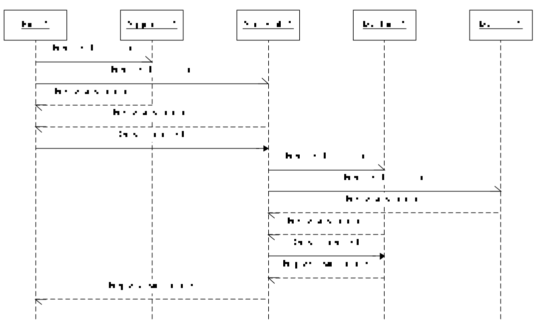

Fig. 13. Contraction based on a simplified Contract Net.

When the requesting messages are sent by agents, information about sending agent and current task conditions enclosed into message. This received information with agents’ beliefs and conditions is used in fuzzy module to determine quality of undertaking the task.

This can help us eliminate from using low quality aggregates and machines even if theirs prime-costs are lower than others. This follows the fact that yield loss could be more than expensive on qualitative cultivation.

When the replies with quality mark and using cost are received contactor chooses the best variant for adding to overall plan and “sign a contract” with all collaborators.

For implementation we are using Jack Intelligent Agents framework [11],[12]. JACK is an important and novel contribution to the field of agent-oriented software engineering. Rather than invent a new language, an existing popular language (Java) has been augmented with constructs for agent communication, knowledge representation, and for both deliberative (goal-based) and reactive (event-based) programming. This has been achieved in a way that allows the programmer to mix familiar Java statements with agent programming constructs. Although JACK is strongly oriented toward the BDI paradigm, its component-based architecture supports a wide variety of agent programming styles. Also JACK architecture allows embedding necessary java modules for extension agent’s possibilities.

5 Related work

It’s supposed to progress development of planning agents in agricultural sector. In particular agent training ability, classification and wide past experience use are in our sight to implement for more intelligent decision making support. Also it is planned to add into system the ability of expert consulting on agricultural machinery selection problem according to local production conditions and market limitations.

Conclusions

The research described in this paper contributes to the multi-agent planning in agriculture sector with distributed knowledge using. This approach opens for farming production new horizons in basic technological planning that was mainly in experts interests.

Our research also has been driven by the idea of embedding the fuzzy logic controller into agent. Transferring knowledge as a set of production rules helps agriculture agents make adequate and soft decisions in changing environment. For this case FCL was adopted with agents.

**The paper is supported of RFBI grants 06-01-00576 and 07-01-00782*

References

[1] Michal Pechoucek, Agent Based Production Planning, PLANET News, 2003.

[2] Rui M. Lima, Rui M. Sousa, Paulo J. Martins, Distributed Production Planning and Control Agent Based System, International Journal of Production Research, 2006, 44(18 - 19): 3693-3709.

[3] Duncan C. McFarlane, Stefan Bussmann Developments in Holonic Production Planning and Control, Int. Journal of Production Planning and Control, Vol. 11, No. 6, 2000, pp. 522 –536.

[4] Skobelev P.O., Multi-agent approach to intelligent system building of coordinated agriculture business managing. 2000.

[5] Jakob Appel, Kurt Nielsen, Applied Multi-Agent Systems Principles to Cooperating Robots in Tomorrow's Agricultural Environment, PhD Thesis, 2005.

[6] V. Novak, I. Perfiliefa, J. Mockor, Mathematical principles of fuzzy logic, Kluwer Academic Publishers, 1999.

[7] D. Rutkovsky, M. Pilnsky, L. Rutkovsky, Neural networks, genetic algorithms and fuzzy systems, Naukowe PWN, 2004.

[8] http://jfuzzylogic.sourceforge.net/

[9] IEC 1131 - PROGRAMMABLE CONTROLLERS Part 7 - Fuzzy Control Programming Committee Draft CD 1.0 (Rel. 19 Jan 97).

[10] Spyridon Fountas, Systems Analysis of Precision Agriculture, PhD thesis, 2004.

[11] Rick Evertsz, Martyn Fletcher, Richard Jones, Jacquie Jarvis, James Brusey, and Sandy Dance. Implementing Industrial Multi-Agent Systems using JACK, 2005.

[12] http://www.agent-software.com/