CHAPTER II[1]

A SIMULATION TOOL FOR MODELING GRAIN STORAGE FACILITIES

Abstract. *This work was carried out with the objective of developing a simulation toolset for performing analyses and evaluations of grain storage facilities that consider dynamic behaviors. This toolset, a library called "Grain Facility," developed using the simulation package Extend^{TM}, has a set of blocks that allows simulation of (i) receiving and dispatching activities, (ii) grain processing unit operations, (iii) operational decisions, and (iv) electrical and heat energy uses. To demonstrate the simulation toolset's application, a grain storage facility that received 73,473.00 t of raw product was modeled. Simulated results show that: (i) after processing unit operations, 12,205.02 t of corn, 47,451.81 t of soybean, and 4,356.88 t of wheat were cleaned and dried, (ii) the maximum amount of product stored occurred in April, when the final monthly balance was 14,409.20 t of soybeans, (iii) the amount of firewood consumed in the drying operation totaled 766.02 t, with 27.0 kg, 13.0 kg, and 16.0 kg of firewood consumed to dry each tonne of corn, soybeans, and wheat respectively, and (iv) that annual electrical energy consumption was 181.98 MWh, 34.78% of which was used in the drying sector.*

Keywords. *grain storage facility, simulation, model, Extend^{TM}*

INTRODUCTION

Several authors have indicated the need for grain drying and storage system simulations. Lower et al. (1994) describes several computer models applied to (*i*) grain drying systems - Thompson et al. (1968), Bridges (1984) and Bridges et al. (1982); (*ii*) facility layout and design - Loewer et al. (1976) and Lower et al. (1990); and (*iii*) total system evaluation - Bridges et al. (1979) and Benok et al. (1981). These models are characterized as deterministic and are applied for system static analysis. Moreover, the models for system evaluation were developed using specific layouts.

[1] Silva, L. C. 2002. Stochastic Simulation of The Dynamic of Grain Storage Facilities. Ph. Dissertation. Universidade Federal de Viçosa. Viçosa, MG. Brazil

According to Flores (1988) and Loewer et al. (1994), a grain storage facility can be defined as a system designed, structured, and equipped for the appropriate receiving, cleaning, drying, storing, and dispatching of grains and legumes. To perform these operations, several types of equipment and structures, such as receiving pits, cleaners, dryers, conveyors, wet holding bins, silos, flat storage containers, and dispatch bins, need to be linked in a logical sequence. It is also required that these systems should be designed, operated, and managed in a way that wanted product characteristics can be preserved and profit can be accomplished.

The operation of a grain storage facility is complex and affected by many factors, some of which demonstrate stochastic behavior that impacts the product-handling process. This type of facility is a high cost, long-term investment. To maximize potential gain from such an investment, structured models that accurately simulate the dynamic behavior of grain storage facilities would be of use. This indicates the need for a simulation toolset to create models that accurately reflect reality, able to consider the influence of stochastic factors and a range of grain facility layouts.

This article was written to present the development of the simulation toolset "*Grain Facility*," which allows the modeling of existing and proposed grain storage facilities.

BACKGROUND

A simulation model can be classified as: (*i*) *static* or *dynamic*, (*ii*) *stochastic or deterministic*, and (*iii*) *discrete* or *continuous*. The static model does not consider the time variable when predicting the state of the system, while the dynamic model does. The stochastic model involves one or more random variables while the deterministic does not. Finally, in the discrete model, time progresses according to event occurrence; in the continuous model, time advances according to a predefined increment, which determines how frequently the model's state variables are recalculated during simulation (Law and Kelton, 1991; Imagine That, Inc., 1997).

The following options can be used to develop a simulation model in digital computers: (*i*) programming languages such as FORTRAN, Pascal, C, or Visual Basic^{TM}, (*ii*) simulation languages such as Arena^{TM}, Extend^{TM}, and SLAM^{TM} or (*iii*) a simulation package such as @Risk^{TM} that works together with a spreadsheet (Roberts and Dessouky, 1998; Maria, 1997; Palisade Corporation, 2000).

THE EXTEND^{TM} SOFTWARE

Extend^{TM} is a graphically oriented simulation package developed by Imagine That, Inc., which can be used for performing discrete and continuous simulation (Krahl, 2000). Users can apply this package in three different ways: (*i*) to run pre-assembled models; (*ii*) to assemble their models using the standard blocks that comes with Extend^{TM}, or (*iii*) to program a new code or modifying the code of existing blocks (Rivera, 1997; Krahl, 2000).

Extend^{TM} models are structured using blocks that are stored in libraries. Each library has a set of blocks that exhibit similar behavior, such as Discrete Event, Generic, Manufacturing, Plotter, and Business Process Reengineering (Rivera, 1997). An Extend^{TM} block can be programmed to model processes, to carry on calculations, to send information, or to plot graphics. It generally is composed of (*i*) an icon that refers to the block's appearance in the worksheet, (*ii*) dialog windows that enable users to define behaviors as well as to access output reports, and (*iii*) connectors that work as interface points and allow blocks to be connected to other blocks (Krahl, 2000).

An Extend^{TM} block can be classified as code or hierarchical. Code blocks are programmed using ModL, Extend^{TM}'s internal programming language. A hierarchical block is assembled using previously constructed blocks. This type of block allows models to be subdivided into sub-models that can represent the system elements. In addition, they enable the user to add documents to the models by nesting text, graphics, and pictures (Imagine That, Inc., 1997; Rivera, 1997).

To build Extend^{TM} models, blocks need to be dragged from the libraries and placed on the worksheet. Next, they need to be logically organized and connected. Through the connections, two types of information flows can be established. The first, exclusively applied to the discrete simulation, refers to items that represent objects or entities that move among the system elements. The second relates to value changes and can be a single number or an array (Rivera, 1997; Krahl, 2000).

Items can have attributes and priorities associated with them. An attribute is a type of variable that describes the item's characteristics. The value of the attribute can be modified during simulation. By assigning priorities the item's importance is specified. This is useful in queue modeling (Krahl, 2000).

Information exchanges among Extend^{TM} blocks can also be established by means of global arrays and global variables. Any block in the model can read the data stored in the global arrays and variables. The user can create global arrays while global variables are predefined by Extend^{TM} (Imagine That, Inc., 1997).

GRAIN FACILITY LIBRARY DEVELOPMENT

The "*Grain Facility"* library was developed using Extend^{TM} software, Version 4.1.3C. This library was developed from the conceptualized model shown schematically in Figure 1. Two types of external stochastic phenomena that affect the grain storage facility system, the grain harvest and the market demand, are assumed to exist and are responsible for the occurrence of the Arrival and Dispatch events. The arrival events refer to the appearance of loaded trucks at the receiving sector. These trucks come from the field, loaded with raw products that must be cleaned and dried. The dispatch events are the arrival of empty trucks at the dispatching sector, which must be loaded with cleaned and dried products. The products in this study are corn, soybeans, and wheat.

The grain storage facility environment, Figure 1, was divided in the following sectors: receiving, wet holding in bins, pre-cleaning, drying, cleaning, storing, and dispatching. These sectors, according to their functions, can have: (*i*) structures, such as receiving pits, wet holding bins, storage bins, flat storage pads, and dispatch bins, and/or (*ii*) processing machines, such as pre-cleaners, cleaners, and dryers. The structures and machines are linked by conveying equipment, such as bucket elevators, screw conveyors, belts, and drag conveyors.

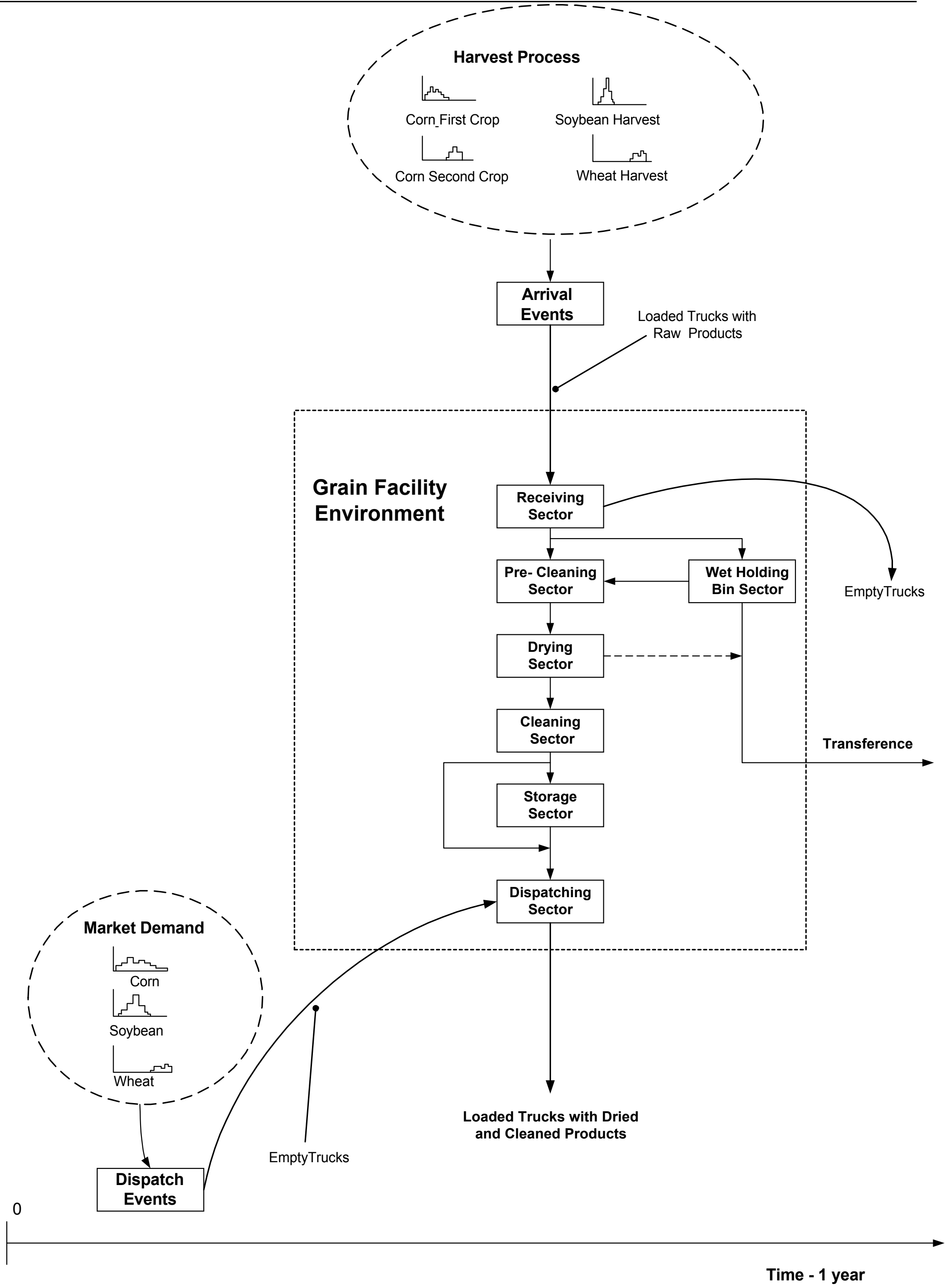


Figure 1 – Representation of the conceptualized model.

The *Grain Facility* library was developed to allow the user: (*i*) to input information that regulates system operations, (*ii*) to simulate unit operations that normally occur at a grain storage facility, (*iii*) to collect and display information during the simulation process, and (*iv*) to generate reports and graphics. To achieve these goals, a set of blocks was developed. The blocks were classified according to type of operation, as shown in Figure 2.

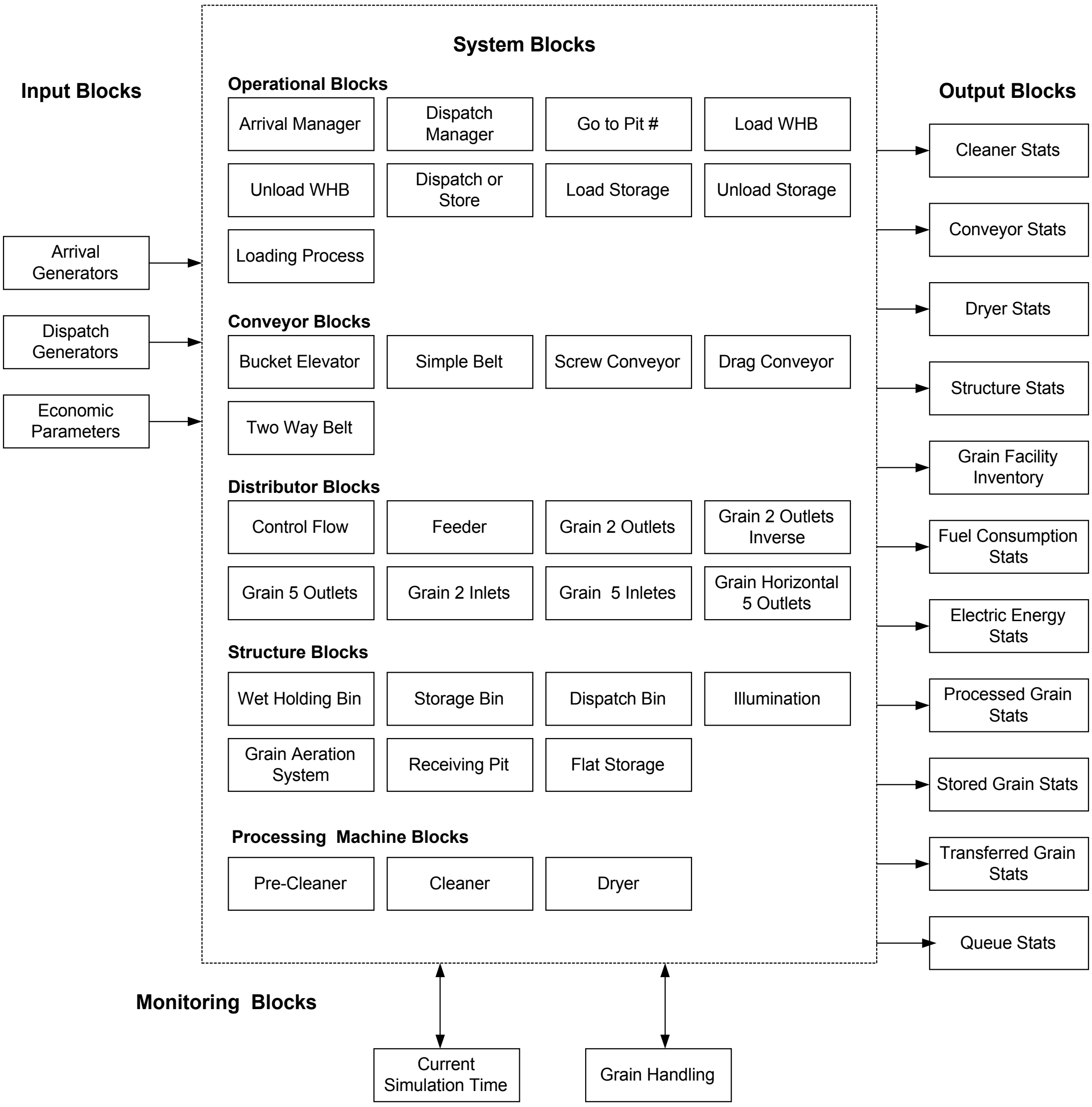


Figure 2 – Grain Facility library organization.

Input Blocks

This category contains the *Arrival Generators*, the *Dispatch Generators,* and the *Economic Parameters* blocks. They define the system operation conditions.

The *Arrival Generators*, a hierarchical block, was developed to generate arrival events. The main dialog window of this block is shown in Figure 3. The events were generated by four sets of blocks. Each set refers to one of the harvesting seasons. In the southern region of Brazil the grain storage facilities generally receive corn grown in two seasons; soybeans and wheat are grown during only one season.

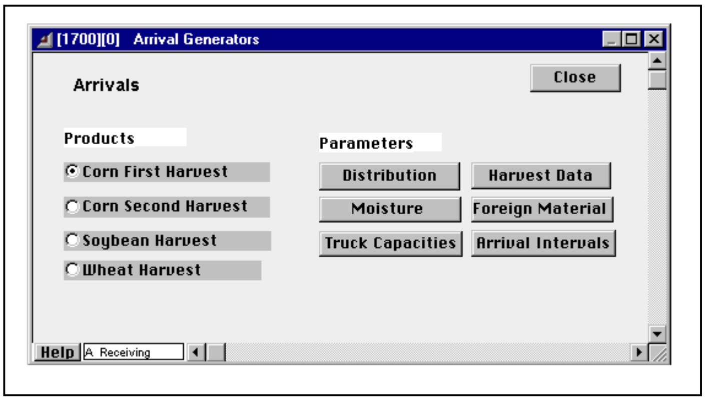
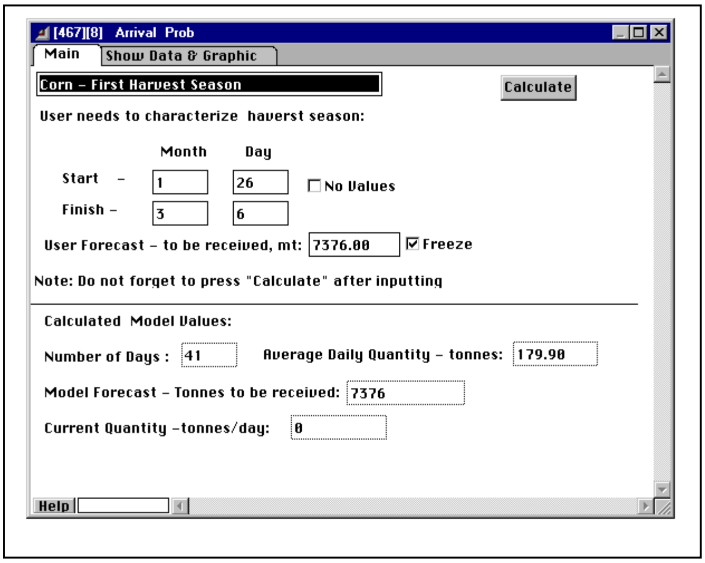


Figure 3 – *Arrival Generators* Block main dialog.

Each set of blocks holds an Extend^{TM} block *Generator*, which was selected to generate items. Each item represents one tonne of raw product. A block called *Arrival Prob* was developed to calculate the number of items to be generated. It was programmed to predict the daily amount of product to be received by the grain storage facility. The prediction was made in accord with user supplied information entered into the model through the dialog window shown in Figure 3. The information refers to distribution and harvest data parameters.

The distribution parameter relates to a fitted or empirical distribution, which describes the variable "*daily harvest success index - DHSI*". This variable defines how many percentage points the day's amount of product received is above or below the average daily value of product received. The average is calculated based on the number of harvest days and the total amount of product to be harvested. This information is introduced in the model through the dialog window shown in Figure 4.

Using empirical or fitted distributions, the user also needs to define other parameters, as shown in Figure 3. These parameters are moisture, foreign material, truck capacity, and arrival intervals. Moisture and foreign material define attributes that are associated to generated items. The truck capacity parameter defines a value that is used by the Extend^{TM} block *Batch (Variable)*. This block allows a defined number of items to be united into a single item, according to a requested number. In this study, the single item obtained represents a loaded truck. The requested number of items to be united refers to the weight of the load, which is also transformed into one more attribute, "*load,*" for a generated item. The parameter arrival interval refers to the time between loaded truck arrivals at a grain storage facility. These empirical or fitted distributions are introduced in the model through the Extend^{TM} block *Input Random Number.*



Corn – First Harvest Season

User needs to characterize havest season:

	Month	Day	
Start –	1	26	□ No Values
Finish –	3	6	

User Forecast – to be received, mt: 7376.00 ☑ Freeze

Note: Do not forget to press "Calculate" after inputting

Calculated Model Values:

Number of Days : 41 Average Daily Quantity – tonnes: 179.90

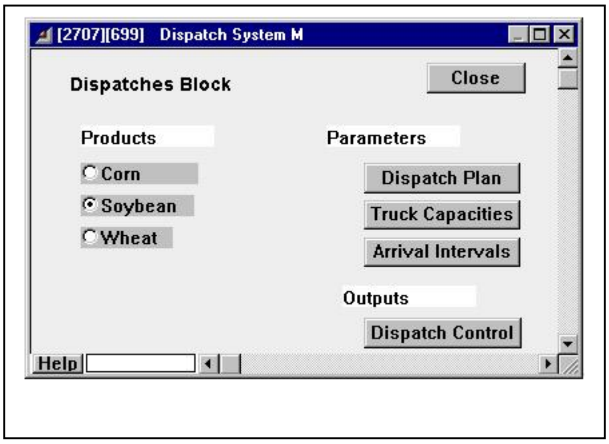
Model Forecast – Tonnes to be received: 7376

Current Quantity –tonnes/day: 0

Figure 4 – Harvest data dialog window.

From each set of blocks responsible for generating arrival events, a group of items representing the traffic of loaded trucks is generated. Each item has five attributes: (i) "*type*" – concerns the type of product: 1 - corn from the first harvest, 2 - soybean, 3 - wheat, and 4 - corn from the second harvest, (ii) "*moisture*" – refers to grain moisture content at arrival, (iii) "*umiCalc*" – also concerns grain load moisture content and is used in the simulated drying process, (iv) "*dust*" – alludes to the load's foreign material content, and (v) "*load*" - alludes to load weight.

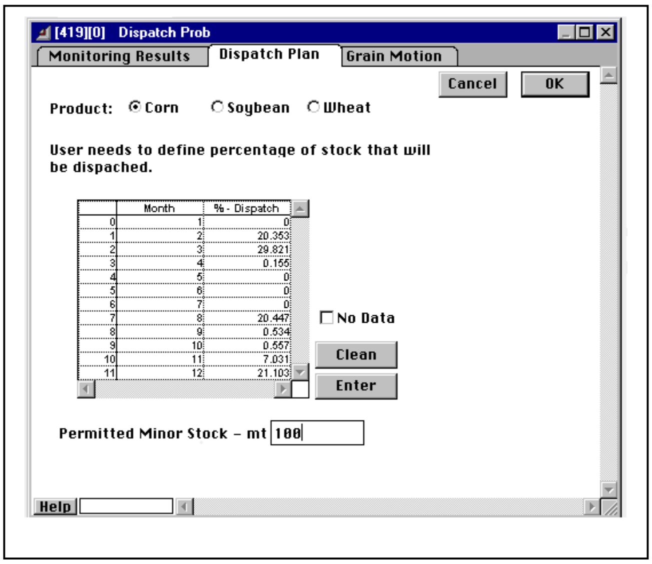
The *Dispatch Generators*, a hierarchical block, generates events related to the arrival of empty trucks at the dispatch sector. The main dialog window of this block is shown in Figure 5. Events are generated by three sets of blocks, since it was assumed that three types of products could be dispatched: corn, soybeans, and wheat. Each set of blocks has the genuine Extend^{TM} block *Generator*. The block *Dispatch Prob* was developed to calculate the number of items to be generated. At the beginning of the simulation, it is used to calculate the conditional probabilities (Kazmier, 1995) according to the parameters of the dispatch plan (Figure 5). This plan refers to the monthly percentage of stock that needs to be delivered, which entered into the model using the dialog window shown in Figure 6. Using conditional probabilities and the model's predicted value for the amount of product to be received, the monthly amount of product that needs to be dispatched is calculated. The information predicted by the model predicted is passed to the *Dispatch Prob* from the *Arrival Prob*, by means of global variable *Global18*.



Products

○ Corn
● Soybean
○ Wheat

Figure 5 - Dispatch event main dialog window.



Product: ● Corn ○ Soybean ○ Wheat

User needs to define percentage of stock that will be dispached.

	Month	% - Dispatch
0	1	0
1	2	20.353
2	3	29.821
3	4	0.155
4	5	0
5	6	0
6	7	0
7	8	20.447
8	9	0.534
9	10	0.557
10	11	7.031
11	12	21.103

□ No Data

Permitted Minor Stock – mt 100

Figure 6 – Dispatch plan dialog window.

During the simulation, the *Dispatch Prob* block is also in charge of calculating the daily amount of product to be dispatched as determined by the monthly amount to be shipped. If current product stock can supply the calculated dispatch demand, this demand value is then passed to the *Generator* block. If supply falls short of demand, the dispatch value is adjusted to the current stock level.

Other parameters to be defined by the user in the *Dispatch Generators* block (Figure 5) are the fitted or empirical distributions related to truck capacity and arrival intervals. The truck capacity parameter defines a value that is used by the Extend^{TM} block *Batch (Variable)* for batching a number of items into a single item. This item represents an empty truck, and the number of clustered items is stored in the attribute called "*DemandedQty."* Thus, each item generated by the *Dispatch Generators* block has two attributes: "*DemandedType"* and "*DemandedQty,"* which respectively refer to the type and amount of the required product.

The *Economic Parameters* block allows the user to specify data that are used in the model for calculating output variables. In this block, the user can access five dialog windows to input the following information: (*i*) monthly electrical energy prices and daily on-peak and off-peak periods, (*ii*) firewood or propane prices, (*iii*) fixed manpower team composition, (*iv*) monthly temporary manpower contracted, and (*v*) management tax, such as installation costs, machine insurance, and central administration tax. "*Global9,*" an Extend^{TM} global variable, passes this information to other blocks.

System Blocks

This category is made up of the blocks used to simulate elements and decision-making processes occurring within the grain storage facility environment (Figure 2). They are grouped in five subcategories: (*i*) operational blocks, (*ii*) conveyors, (*iii*) distributors, (*iv*) structure, and (*v*) processing machines.

Operational Blocks

This category has blocks that were developed to simulate management decision-making in grain storage facilities and to specify the constraints that establish the system logic. Figure 7 shows the adopted operational flowchart employed for modeling operational activities. Ten types of decisions were considered.

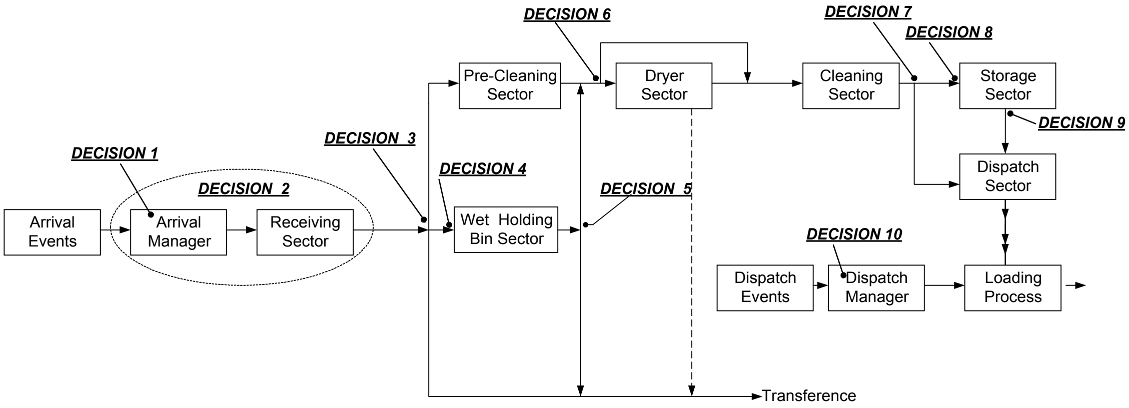


Figure 7 – Schematic operational flowchart decisions

- Decision 1 - the trucks are separated in queues according to product type. This decision was modeled using the hierarchical block called *Arrival Manager*.
- Decision 2 – defines to which receiving pit the loaded truck should go. The *Go to Pit#* block inside the *Arrival Manager* block defines this decision.
- Decision 3 - defines to which sector of the system the product from the receiving pit should go. The options are to the pre-cleaning sector, the wet holding bin sectors, or to be transferred to another grain storage facility. Transference to another grain storage facility occurs when the raw product is held at the receiving pit more than a specified period of time as defined by the user. The product is sent to pre-cleaning when the dryer can receive it. If pre-cleaning cannot receive the product, it is transferred to the wet holding bin sector. Decision 3 is modeled by the block *Unload Pit*.

- Decision 4 - defines in which holding bin the product should be stored. This decision is based on the type of conveyed product and type of product stored in the wet holding bins. The *Load WHB* block simulates this decision.

- Decision 5 - defines to which sector the product leaving the wet holding bin should go. The options are to go to pre-cleaning/drying sectors or to be transferred to another grain storage facility. Transference to another grain storage facility occurs when the raw product is held at the wet holding bin sector more than a specified period of time. The *Unload WHB* block simulates this decision.

- Decision 6 - defines if the product needs to be passed through the dryer sector. If the product has moisture content below 14.9% w.b., it is conveyed to the cleaning sector.

- Decision 7 – defines the direction of grain flow leaving the cleaning sector. The product can go to the storage sector or to the dispatch sector. If the type of product leaving the cleaning sector is required by the dispatch sector, the product goes to this sector. This decision is made by *Dispatch or Store* block.

- Decision 8 - defines to which bin or flat storage cell the product should go. This decision is based on the type of the conveyed product and the product that is in storage. The *Load Storage* block simulates this decision.

- Decision 9 - defines from which bin or flat storage cell the product should be unloaded. This decision is based on the product required by the dispatch sector and the amount of product stored in each bin or flat storage cell. The one that has the least product is unloaded first. The block used to model this decision is the *Unload Storage* block.

- Decision 10 - defines to which queue the empty truck should go. This is decision is determined by the type of product to be loaded in the truck. If there is no product available, the truck returns empty. Otherwise, the truck loading operation is performed. The block used to model decision 10 is *Dispatch Manager*.

The *Loading Process* block simulates the truck loading operation. It receives items that come from two sources: *Dispatch Generators* and processing or storage sectors blocks. Items from *Dispatch Generator* block represent empty trucks and have two attributes: "*DemandedType*" and "*DemandedQty*". Each item held in processing or storage sectors blocks represents one tonne of product. The Extend^{TM} block called *Batch* was used in the *Loading Process* block to allow items from the processing and storage sectors to be grouped with items from *Dispatch Generator*. Thus, a single item is formed, which represents a loaded truck.

All needed information to support the simulation of decision-making by grain storage facility management is passed among blocks by means of value connectors, global arrays, or global variables. In the modeling, fourteen Extend^{TM} global variables were employed and one global array called "*GrainMonitor*" was created.

Distributors

Seven blocks are included in this category: *Control Flow*, *Feeder*, *Grain 2 Outlets*, *Grain 2 Outlets - Inverse*, *Grain 2 Inlets*, *Grain 5 Inlets,* and *Grain Horizontal 5 Outlets. Control Flow* is the only hierarchical block and was developed to govern the amount of items that need to be conveyed to the *Dispatch Bin* block. This amount is defined according to the value of the attribute "*DemandedQty."* The six other blocks in this category were developed using Extend^{TM} *routing* blocks from the Discreet Event and Manufacturing libraries and have had their icon appearances modified, as shown in Figure 8, to ease their identification in the grain storage facility models.

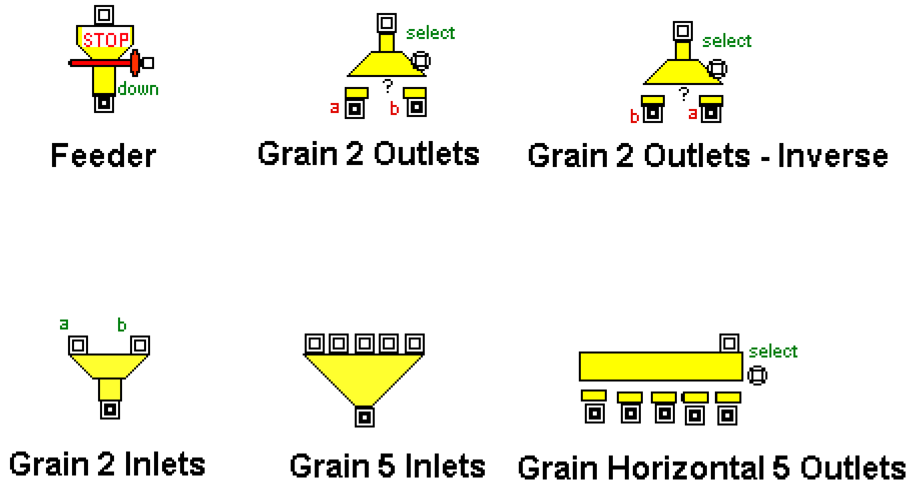


Figure 8 – Distributor blocks.

Conveyors

In this category, the following blocks were created: *Bucket Elevator*, *Simple Belt*, *Screw Conveyor*, *Drag Conveyor*, and *Two Way Belt*. The first three are code blocks that only differ in icon appearance. They were developed using the Extend^{TM} block called *Machine* contained in the Manufacturing Library. The following modifications were applied: (*i*) the block icon appearance was changed to better represent bucket elevators, screw conveyors, and belt conveyors, (*ii*) the variable *processing time* was defined to be calculated according to the hourly capacity of conveyors (t/h), (*iii*) the variable *processing time* was defined to be updated during the simulation according to the attribute "*realvalue*". This attribute was defined to express the update mass of each tonne of product that arrived in the grain storage facility and (*iv*) computational routines were introduced to calculate electrical energy use.

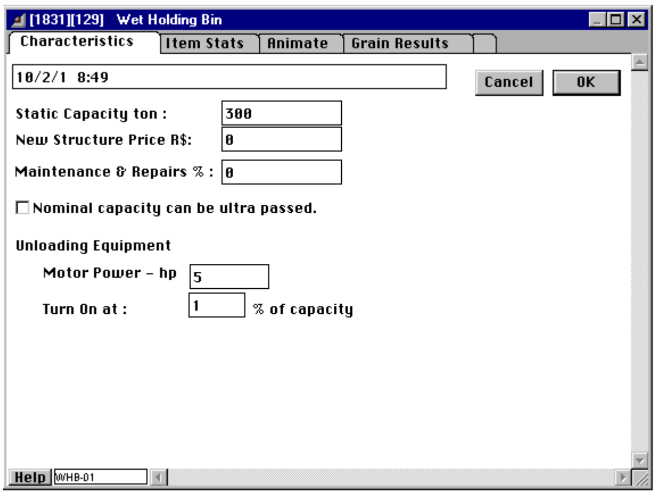
Electrical energy use is calculated according to the electrical power requirement of the equipment and utilization time. The results obtained refer to consumption and demand, expressed in kilowatt-hours (kWh) and kilowatts (kW) respectively. These results are calculated for on-peak and off-peak hours. The user provides this information for simulation through the *Economic Parameters* block.

Drag Conveyor and *Two Way Belt* are hierarchical blocks, which have the *Simple Belt* as the main block. *Drag Conveyor* uses two blocks: *Simple Belt* and *Grain Horizontal 5 Outlets*. The *Drag Conveyor* block simulates the conveyor operation, and the second, the flow distribution in accordance with the number of linked exits. The *Two Way Belt* block has two *Simple Belt* blocks, one of which simulates the forward direction of the product conveyed; the other simulates the backward direction.

Structure Blocks

This category is made up of the following blocks: *Wet Holding Bin*, *Storage Bin*, *Dispatch Bin*, *Illumination, Grain Aeration System*, *Receiving Pit,* and *Flat Storage*. The first three simulate specific types of bins and have the same codes, only differing in icon appearance. They were developed using the Extend^{TM} block called *Buffers*, contained in the Manufacturing Library. The following modifications were implemented: (*i*) the icon block appearance was changed for a better representation of the wet holding bin, the storage bin, and the dispatch bin. Three routines were developed; the first routine calculates the amount of product stored according to the values of the attributes *"type"* and *"realValue"*. The second routine reports information about the type and amount of product, and global variable "*GrainMonito*r" passes this information to other blocks. The third routine calculates the bin unloading system's electrical energy consumption.

Figures 8 and 9 show the major structure blocks dialog windows. The dialog window in Figure 8 is used for characterizing the facility's structure, while the dialog window shown in Figure 9 is used to access the simulation results. When working with the Wet *Holding Bin*, *Storage Bin*, *and Dispatch Bin* blocks, the user must provide the unloaded system's electrical power rating and the amount of product already stored in the bins when the unloading equipment starts to work.



10/2/1 8:49

Static Capacity ton : 300

New Structure Price R$: 0

Maintenance & Repairs % : 0

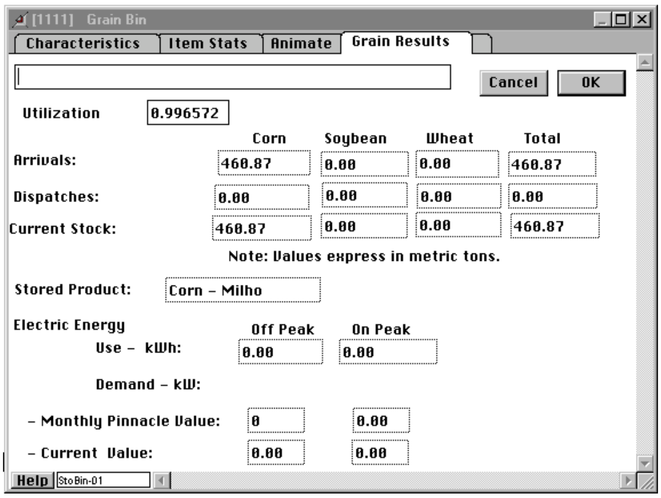
□ Nominal capacity can be ultra passed.

Unloading Equipment

Motor Power – hp 5

Turn On at : 1 % of capacity

Figure 8 – Main dialog window of structure blocks.



Utilization 0.996572

	Corn	Soybean	Wheat	Total
Arrivals:	460.87	0.00	0.00	460.87
Dispatches:	0.00	0.00	0.00	0.00
Current Stock:	460.87	0.00	0.00	460.87

Note: Values express in metric tons.

Stored Product: Corn – Milho

Electric Energy	Off Peak	On Peak
Use – kWh:	0.00	0.00
Demand – kW:		
– Monthly Pinnacle Value:	0	0.00
– Current Value:	0.00	0.00

Figure 9 - Results dialog window of the structure blocks.

The *Illumination* and *Grain Aeration System* blocks were developed to simulate the electrical energy use of the offices and for external illumination and for the storage aeration system's the fan operations. For both blocks, the user needs to define the electrical power rating of the equipment and the equipment's run cycle times. Thus, electrical energy use is calculated during the simulation.

The hierarchical block *Receiving Pit* was developed to simulate two processes: the truck unloading operation and the storage of raw products. In the modeling, the *Simple Belt* block was employed for simulating the first process, and the *Storage Bin* block for the second. The user needs to specify a fitted or empirical distribution that describes the time spent unloading and the storage capacity of the receiving pit. Also, if the receiving pit is equipped with a mechanical truck unloading system, the user needs to input the electrical power used by the motor that makes this system work. The developed block *Motor* simulates this electric energy consumption.

The *OperationPit* block nested in the *Receiving Pit* block generates a protocol that describes the simulated operational condition of a receiving pit. The protocol has six digits "*TUVWXY*" that refer to: (*i*) "*T"* - the number associated with the receiving pit for identification purposes, for example 1, 2, 3, or 4; (*ii*) "*U"* - type of product in the receiving pit: 1: corn, 2: soybean, and 3: wheat; (*iii*) "*V"* - moisture content of the product: 0 if it is up to 18% w.b. and 1, if it is above 18% w.b.; (*iv*) "*W"* - 1 if the pit bay is busy, 0 if not; (*v*) "*X"* - 1 if pit is full, 0 if not; and (*vi*) "*Y"* – 1 if the item is held in the last *Buffer* block of the *Receiving Pit*, 0 if otherwise 0. The *Buffer* is an Extend^{TM} block. Thus, each of the model's *Receiving Pit* blocks generates a protocol. The analyses of these protocols support decisions 2, 3, and 4 (Figure 7).

As it was previously mentioned in the *Arrival Generators* block description, the generated items are grouped into a single item, which represents a loaded truck. The requested number of items to be grouped is stored in the attribute "*load."* This grouped item is separated in the *Receiving Pit* block.

The *Receiving Pit* block uses the Extend^{TM} block called *Unbatch (Variable)*. The *Unbatch (Variable)* block allows a single item to be turned into a number of identical items. This way, the *Unbatch (Variable)* is used to separate the items that were grouped in the *Arrival Generators* blocks. The number of items resulting from this separation is defined according to the value of the attribute "*load."* After the unbatching process, each item represents one tonne of product and receives three more attributes: (i) "*realvalue*" – used for updating the mass of one item according to simulation of unit operations. The first value of this attribute is 1, as it refers to one tonne of raw product, (ii) "*intermittent*" -

used in the *Dryer* block for defining the need for the product to re-circulate through the dryer, (*iii*) "*empty*" - used in the *Dryer* block for defining in which situation this block can be emptied.

In Brazil, flat storage structures are divided in cells that allow storing products of different types. To consider each internal cell working as a single storage unit, the hierarchical block *Flat Storage* was developed. Since flat storage structures generally contain a maximum of three cells, *Flat Storage* was modeled using three *Storage Bin* blocks.

Processing Machines

This category includes the *Pre-Cleaner*, *Cleaner,* and *Dryer* blocks. The first two blocks have the same code and differ only in icon appearance. They were developed using the *Machine* block from the Extend^{TM} Manufacturing Library. The modifications applied were the same as those adopted for modeling conveyors with the addition of one more routine for simulating the cleaning operation. This routine: (*i*) adjusts the value of the attribute *"realValue*" according to initial and final foreign material content, (*ii*) updates the value of the attribute "*dust,*" and (*iii*) calculates the amount of the foreign material removed.

The initial foreign material content refers to the value of the attribute *"dust*" when the item arrives at the *Pre-Cleaner* or *Cleaner* blocks. The user needs to define machine efficiency according to: (i) moisture content of product, and (ii) the possible reduction of foreign material content. The first definition affects the hourly capacity of the equipment while the second defines the amount of foreign material to be removed. In the modeling process, data presented by Weber (2001) was used.

The *Dryer* block, classified as hierarchical, was designed to simulate the drying process in mixed flow dryers (Brooker et al., 1992) by employing the discrete simulation modeling concept. Thus, a dryer was conceptualized as a workstation that applies drying treatment to a defined number of items, which is determined by the dryer's static capacity.

Figure 10 shows a schematic representation of the *Dryer* block. The rectangular and diamond boxes refer to procedures and decisions respectively. Procedures and decisions were modeled using specific sets of blocks.

The Initial Stats procedure, Figure 10, was modeled to compute the input flow of wet product. When the item attribute "*intermittent*" has the value –1, the item has been inputted only once at the *Dryer* block. This way, a block called *Flow* is in charge of computing this input, according to the item attributes "*type*" and "*realValue.*" The attribute "*intermittent*" can also have its value of –1 changed to 1. This happens if the cooling section of the dryer tower is full. The *Dryer Tower* block, described below, reports this information.

The Tower procedure (Figure 10) relates to the physical structure of the dryer that holds the product during the drying process. The *Dryer Tower* block is in charge of simulating this structure. This block simulates a first-in-first-out queue and was developed from the Extend^{TM} block *Buffer*. The allowed maximum length of the queue represents the dryer's static capacity (t).

In the dialog window of the *Dryer* block, the user needs either to input the dryer's static capacity or accept the model suggestion. This suggestion is indicated according to values cited by Weber (2001). One third of a dryer's static capacity corresponds to the cooling section's static capacity. Also, the user has to input the minimum amount of product needed to begin the dryer's operation. This information defines the minimum number of items that need to be in the *Dryer Tower* block to simulate the dryer's operation. The model also offers a suggestion of this value.

During simulation, *Dryer Tower* block reports the number of held items as well as the averages of their attributes "*umiCalc*" and "*moisture.*" Attributes "*umiCalc*" and "*moisture*" refer to the current and the initial product moisture content respectively.

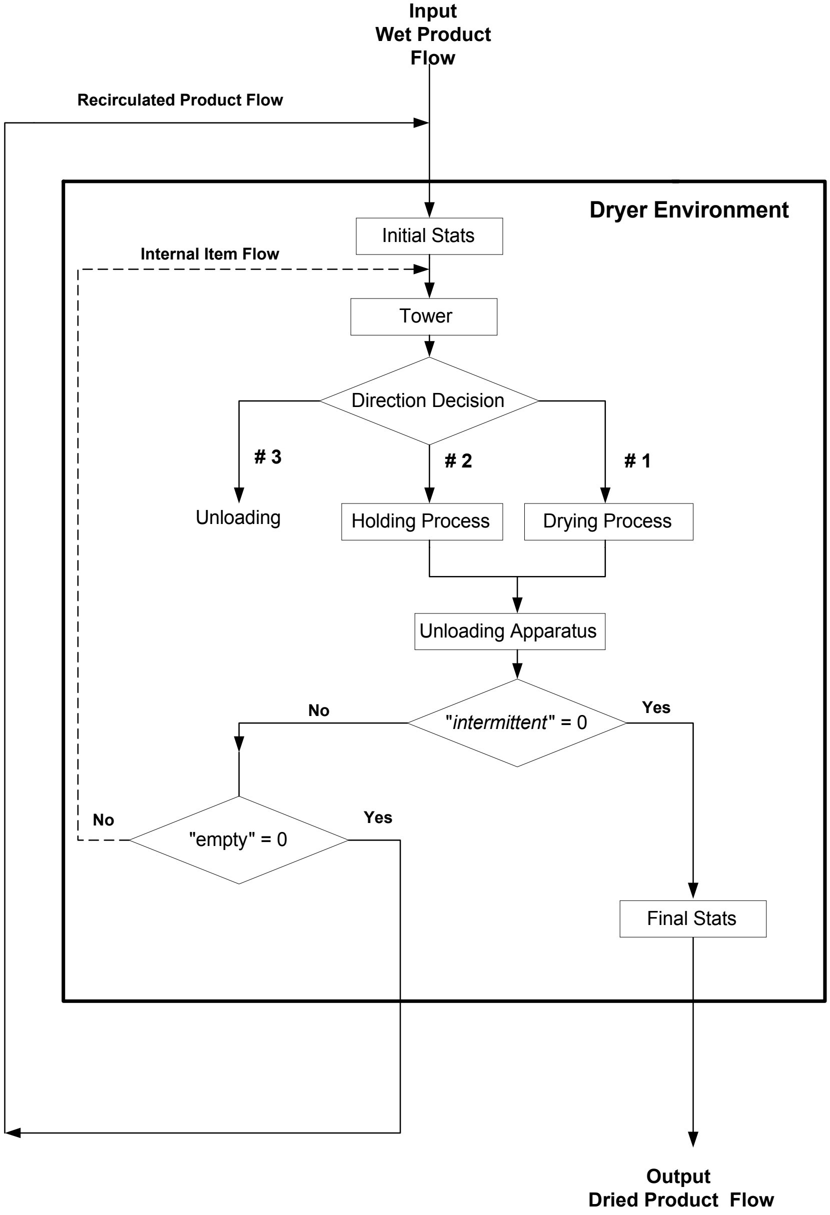


Figure 10 – The diagram representation of *Dryer* block.

From the Direction Decision (Figure 10), items can be sent in directions #1, #2 or # 3. Direction #1 is defined if *Drye*r *Tower* block reports: (*i*) an "*umiCalc*" average that is above the desired final moisture content, and (*ii*) the number of held items is above the minimum number of needed items for simulating the dryer's operation. In this case, the procedure Drying Process is executed. This procedure simulates: (*i*) the dryer fans' electrical energy consumption, (*ii*) the decrease in the product's moisture content, (*iii*) fuel consumption, and (*iv*) the need for the product (item) to be re-circulated.

- The electrical power rating of the dryer's fans is determined using the "*Simple Belt*" block, which calculates consumption according to that rating and the fan's time in use. According to the behavior of the "*Simple Belt"* block, the user needs to input the dryer capacity (t/h) and electrical power rating (kW) of the fans.

- The simulation of the decrease in product moisture content involves a set of blocks that deal with two item attributes: "*umiCalc*", and "*intermittent*". The attribute "*umiCalc*" has a value equal to the initial moisture content when the item enters the *Dryer* block for the first time. After that, each time that an item is submitted to the Drying Process procedures (Figure 10), the value of its attribute "*umiCalc*" can be decreased. The decreased value is defined according to an empirical distribution shown in Figure 11. This distribution correlates the value of initial moisture content with percentage points the moisture content is reduced. During simulation, the *Dryer Tower* block reports the average of the held item's "*moisture*" attributes. This value defines the reduction that needs to be applied to the attribute "*umiCalc*". The value of "*umiCalc*" is only reduced if "*intermittent*" is equal to 1. The empirical distribution shown in Figure 11 was defined according to: (*i*) the mixed flow dryer efficiency presented in Figure 12 (Weber, 2001), (*ii*) the initial moisture content of product, and (*iii*) the dryer's hourly capacity (t/h). With this information, it is possible to calculate the number of times that the product load needs to be re-circulated within the dryer to achieve the desired final moisture content.

- Fuel consumption is calculated considering: (i) the number of items that pass through Direction #1 (Figure 10) and had their attribute "*umiCalc"* decreased, (ii) the dryer capacity (t/h), and (iii) the hourly fuel consumption of the heating system. The user needs to define this hourly fuel consumption or adopt the model's suggestion. This suggestion is formulated considering dryer capacity (t/h) and the dryer specific heat energy consumption, expressed in kJ per kg of evaporated water. This last information is suggested by the *Dryer* block in accord with filed data, manufactures' catalogues, and data from by Weber (2001).

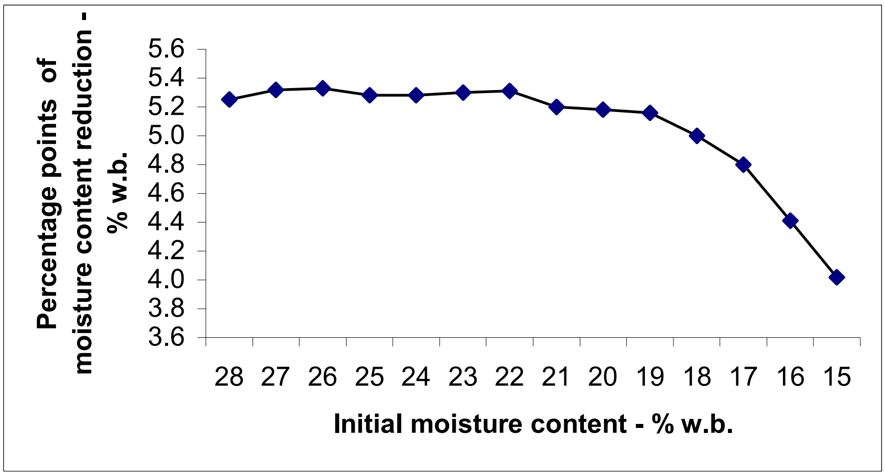


Figure 11– Empirical distribution for moisture content drop in function of initial moisture content.

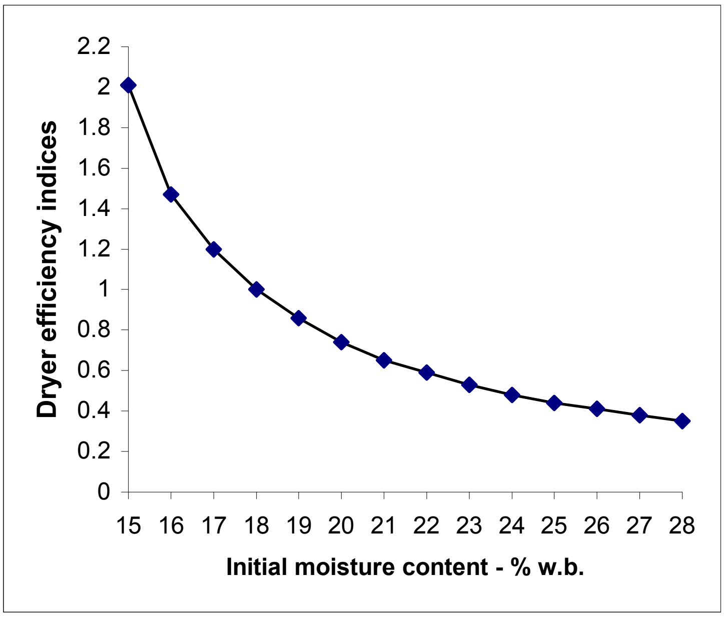


Figure 12 – Mixed flow dryer efficiency indices as a function of initial moisture content considering a final moisture content of 13.0% w. b. (Weber 2001).

- The need for product (item) to be re-circulated through the *Dryer* block is defined according to the values of "*intermittent*" and "*empty.*" In the Drying Process procedure (Figure 10), the attribute "*intermittent*" receives the value 1 if item needs to be re-circulated through *Dryer* block. This happens if the *Dryer Tower* block reports that the held item's average "*umiCalc*" attribute is: (i) higher than 18.3, or (ii) lower than 14.2, assuming that there is the minimum amount of product (item) needed for dryer operation to occur. The attribute "*empty*" was adopted to guarantee the complete drying simulation of the last product load in the dryer. "*Empty"* receives the value 1 if the *Dryer Tower* block reports that the held item's average "*umiCalc*" attribute is below 14.66; otherwise it receives the value 0.

Direction # 2 (Figure 10) is defined if the *Dryer Tower* block reports the of held item's average "*umiCalc*" attributes is lower than or equal to the desired final moisture content, and the amount of held item is equal or lower than the number of items needed for the simulation of dryer operation. If the user had defined that dried product can be held for a given time waiting for a new load of wet product, the Holding Procedure will shutdown the flow of items. The flow will be reestablished if the waiting period is exceeded or a new product load (items) arrives at the *Dryer* block.

Direction #3 (Figure 10) is proposed when the average "*umiCalc"* attribute for the item held in the *Dryer Tower* signals that the it has a higher than desired final moisture content, but the number of held items does not allow simulating the dryer operation. When this situation occurs, the *Dryer* block should be emptied.

Items that go to directions #1 and #2 necessarily are submitted to the *Unloading Apparatus* procedure. This procedure is in charge of simulating the electrical energy use of the dryer unloading system. The "*Simple Belt"* block was employed for simulating this energy consumption. The user needs to input the dryer's handling capacity (t/h) and electrical power rating of the dryer unloading system.

After *Unloading Apparatus* procedure, items are submitted a decision test involving the value of the attribute "*intermittent*" (Figure 10). If the attribute value is 0, the item goes to the *Final Stats* procedure; otherwise, the item is submitted to a new decision involving the attribute "*empty.*" If the value of "*empty*" is 0, the item goes through the *Dryer* block again. In this case, the item is externally conveyed to the *Dryer* block's input area using one of the conveyor blocks. If the value of "*empty*" is 1, the item is internally conveyed to the *Dryer*

Tower. This was a modeling artifice used to guarantee the drying simulation of the last load of wet product. As it was earlier explained, the *Dryer* block needs a defined minimum number of held items in the *Dryer Tower* block to simulate the drying process.

The goals of the Final Stats procedure (Figure 10) are to: (*i*) to calculate the amount of water removed during the drying process for each item. This calculated value is then stored in the attribute "*waterRemoved*"; (ii) to update the value of the attribute "*realValue"* for each item according to the amount of water removed; and (*iii*) to register the total quantities of dried product and water removed as a result of the simulated drying.

The *Dryer* block generates a protocol that is used by decisions 3 and 4 (Figure 7). The generated protocol has seven digits, "*KLMNOPQ.*" These digits refer to: (*i*) "*K"* - the number associated to the dryer identification, for example 1, 2, 3, or 4; (*ii*) "*L"* - type of product in the dryer: 1: corn, 2: soybean and 3: wheat; (*iii*) "*M"* - if the dryer is full 1, otherwise 0; (*iv*) "*NOP"* - static dryer capacity; and (*v*) "*Q"* - if the dried product is held waiting for a new load of wet product 1, otherwise 0.

Monitoring Blocks

The blocks included in this category were developed for calculating, collecting, and passing the information needed during simulation. The *Current Simulation Time* block, for example, was developed for calculating the time variables year, month, day, hour, minute and weekday, and passing these values to the others blocks in the model. The values of these variables are defined according to the value of the Extend^{TM} system variable "*CurrentTime."*

The *Grain Handling* block was developed to collect information about the *Dryer* block and all blocks categorized as structure blocks. These types of blocks access the created global array called "*GrainMonitor*" to report information related to: (*i*) system element behavior, such as type and static capacity, and (*ii*) held items, such as the number and type of held items and the sum of these held items *"realValue"* attributes*,* which refers to the amount of the product being held. The information stored in the components of *"GrainMonitor*" is processed in the *Grain Handling* block and passed to the Operational

Decision Blocks. This information is passed through the Extend^{TM} global variables.

Output Blocks

The blocks of this category (Figure 2) were developed using *"Activity Stats"*, an Extend^{TM} block of the Stats library, and are intended to: (*i*) get information from specific blocks according to a frequency defined by the user, (*ii*) process this information, and (*iii*) elaborate reports in table or graphic format.

The *Electric Energy Stats* output block, which simulates electrical energy metering equipment, is in charge of collecting the output data from blocks that simulate electrical energy use. These data refer to the consumption (kWh) and demand (kW), according to on and off-peak hours. In this way, it is possible to obtain information for calculating monthly electricity bills from the *Electric Energy Stats* block's records.

APPLICATION OF THE GRAIN FACILITY LIBRARY

The Grain Facility library was used to model a grain storage facility belonging to COAMO, an agricultural cooperative of grain producers headquartered in Campo Mourão, Paraná, Brazil. Details about the equipment and structures used at this grain facility are presented in Table 1.

Table 1 – Some technical information of the modeled grain storage facility

Structure	**Quantity**	**Static capacity (t)**
Receiving Pit	4	300
Wet Holding Bin	2	300
Flat Storage	1	
Cell-01		5,000
Cell-02		8,000
Cell-03		5,000
Processing machines	**Quantity**	**Hourly capacity (t/h)**
Pre-Cleaner	3	40
Dryer	1	80
Cleaner	4	30
Conveyors	**Quantity**	**Hourly capacity (t/h)**
Belt	4	120
Two way belt	1	120
Bucket elevator	7	120
Drag conveyor	2	120

The following information was collected to build the model: (i) data about structures, processing machines, and conveyors, such as type, static capacity (t), handling capacity (t/h), and electrical power rating (kW); (*ii*) the grain storage facility operational flowchart, (*iii*) data from the 1999 harvest season (Table 2); and (*iv*) data regarding the facility's dispatch plan (Table 3).

MODEL OUTPUTS AND DISCUSSION

Since the developed model is stochastic, each run refers to different sets of inputs and outputs. To illustrate types of model information output, results from one replication in which the simulation time was 8640 hours (12 months) are presented. Figure 13 graphically shows the monthly amount of products received, processed, stored, and dispatched; and Table 4 presents the summarized model output. The amount received information was collected at the *Arrival Manager* block, and represents the amount of product that was unloaded at the receiving sector. These data were generated according to the information shown in Table 2. The model simulated a combined input of 73,473.00 t of corn, soybeans, and wheat. However 4,516.33 t were transferred to other grain storage facilities. April was the month with maximum number of trucks arrivals 1,539 trucks, corresponding to 22,904.00 t of product unloaded.

The processed quantities of product are the values in tonnes obtained after product pre-cleaning, drying, and cleaning operations. It was verified that 3,703.43 t of water, and 1,239.56 t of foreign materials were removed. Thus 12,205.02, 47,451.81, and 4,356.88 tonnes of dried and cleaned corn, soybean and wheat were obtained respectively (Table 4).

Table 2 – Information used at the *Arrival Generators* block

Product	**Distribution types for predicting the "*daily harvest success index" - DHSI***	**Received amount of products (t)**	**Harvest period**	**Moisture content**		**Foreign material content**	
				% w.b.	**% of loads**	**%**	**% of loads**
Corn	Beta	7,376	01/26/99	lower than 14.2	1.42	1.1 to 2.0	93.87
(1^{st} Crop)	$\propto_1$ = 0.2287 $\propto_2$ = 0.5360		to	14.3 to 18.2	4.22	2.1 to 3	6.13
	Minimum = 0 Maximum = 317.18		03/06/99	18.3 to 24.0	57.60		
				higher than 24.0	36.75		
Corn	Exponential	7,804	07/17/99	lower than 14.2	1.64	lower than	3.37
(2^{nd} Crop)	β = 101. 355		to	14.3 to 18.2	10.27	1.0	96.63
	Shift = -1.3697		09/30/99	18.3 to 24.0	55.12	1.1 to 2.0	
				higher than 24.0	33.96		
Soybean	Beta	53,179	02/26/99	lower than 14.2	55.02	1.1 to 2.0	92.54
	$\propto_1$ = 0.3665 $\propto_2$ = 1.7942		to	14.3 to 18.2	39.03	2.1 to 3.0	6.88
	Minimum = 0 Maximum = 686.70		04/30/99	18.3 to 24.0	5.85	3.1 to 6.0	0.56
Wheat	Exponential	5,118t	08/16/99	14.3 to 18.2	54.60	lower than	29.22
	β = 100.001		to	18.3 to 24.0	45.40	1.0	54.55
	Shift = -2.2728		09/29/99			1.1 to 2.0	16.24
						2.1 to 3.0	

Table 3 – Dispatch plans for corn, soybean and wheat

Month	Stock portions to be dispatched (%)		
	Corn	Soybean	Wheat
1	0	0	0
2	20.35	0	0
3	29.82	47.18	0
4	0.15	14.52	0
5	0	20.58	0
6	0	0.67	0
7	0	6.52	0
8	20.45	7.81	0
9	0.55	0.54	69.96
10	0.55	0	0
11	7.03	1.32	0
12	21.10	0.86	30.04

Stored quantities refer to the amount held in the storage sector at the end of the month. According to the simulated scenario, it can be verified that 52,868.81 tonnes were stored over the year, distributed in the following manner: Cell-01 18,107.17 t; Cell-02 9,167.31 t, and Cell-03 25,594.33 t. Cell-01, Cell-02, and Cell-03 also presented rotation indices equivalent to 3.62, 1.15, and 5.11 respectively, and time in use of 10.60, 1.16, and 7.97 months respectively. According to Figure 13, the greatest amount of product stored occurred in April, when all Flat Storage cells were filled with soybean and the final stock for this month was 14,409.20 t. As shown above, in some months two types of products are simultaneously stored.

The dispatched product quantities were delivered according to the dispatch plans shown in Table 3. The greatest monthly quantity dispatched was in March, 27,809.43 t, which corresponded to 1,219 delivered loads with an average weight of 22.81 t.

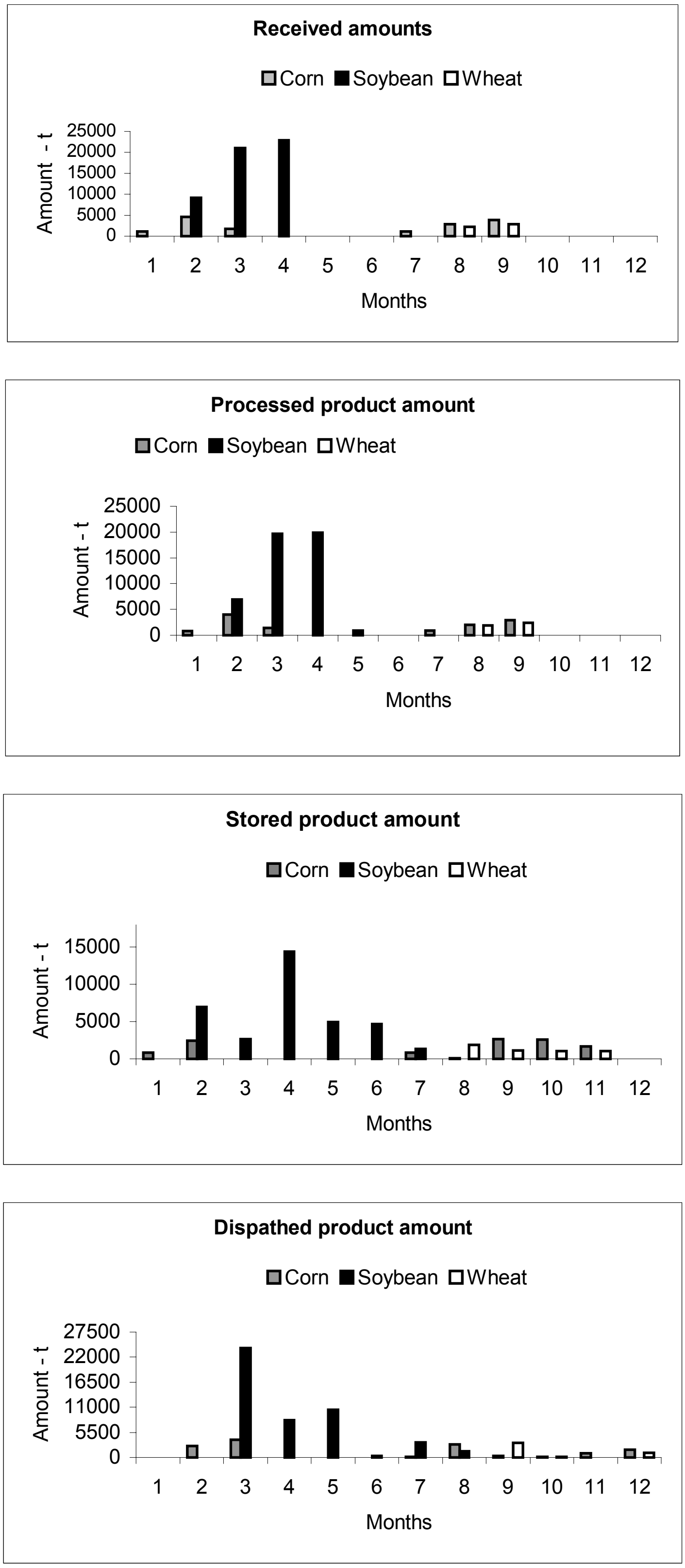


Figure 13 – Monthly quantities of corn, soybeans, and wheat that were received, processed, stored, and dispatched.

Table 4 – Data related to the electric energy use and product flows among the grain storage facility's sectors

Sector (Installed Power)	Electric energy consumption (MWh)	Amount of handled products (tonnes)				
		Movement	Corn	Soybean	Wheat	Total
Receiving (50 kW)	19.9	Input	15,176.00	53,179.00	5,118.00	73,473.00
		Transferred	335.00	2,288.00	0	2,623.00
		Output	14,841.00	50.891.00	5,118.00	70.850.00
Pre-Cleaning (20.9 kW)	12.9	Input	14,841.00	50.891.00	5,118.00	70.850.00
		Output	14,638.18	50,165.55	5054.75	69,858.45
Drying (87.3 kW)	111.5	Input	14,226.92	23,031.28	5,042.81	42,301.00
		Recirculation	37,419.16	31,329.80	8,899.01	77,647.97
		Transferred	551.77	1,087.32	254.23	1,893.32
		Output	11,840.14	20,503.79	4,360.34	36,704.26
Cleaning (33.6 kW)	36.1	Input	12251.38	47,638.06	4,372.28	64,261.72
		Output	12,205.02	47.451.81	4,356.88	64,013.71
Storage (52.2 kW)	28.9	Input	11,813.51	36,994.62	4,060.69	52,868.81
		Output	11,813.51	36,994.62	4,060.69	52,868.81
Dispatch (22.4 kW)	10,9	Input	12,205.02	47,451.81	4,356.88	64,013.71
		Output	12,205.02	47,451.81	4,356.88	64,013.71
General Use (1.5 kW)	6.3	-	-	-	-	-
Total	226.6					

Table 5 shows the simulated firewood and electrical energy use information. Over the simulated year, firewood consumption was 766.02 tonnes, while specific uses were 27.0, 13.0, and 16.0 kg per tonne of dried corn, soybeans, and wheat respectively. According to simulated output data, 1,835.02, 1,440.17, and 428.23 tonnes of water were removed from the received quantities of corn, soybean, and wheat respectively.

Table 5 – Firewood and electric energy use data

Month	Firewood consumption (t)			Electric energy use			
				Consumption (kWh)		Demand (kW)	
	Corn	Soybean	Wheat	Off peak	On peak	Off peak	On peak
1	26.34	0.00	0.00	3012.60	558.97	185.81	178.73
2	128.82	38.02	0.00	25961.86	3608.49	264.86	182.27
3	49.06	127.96	0.00	44081.03	9372.16	266.06	233.08
4	0.00	124.89	0.00	40008.93	4538.77	266.06	215.36
5	0.00	5.96	0.00	5661.40	196.26	201.19	27.23
6	0.00	0.00	0.00	374.60	45.00	22.76	0.30
7	26.49	0.00	0.00	3792.25	332.30	208.27	178.73
8	62.50	0.00	34.09	14317.73	1356.06	236.62	207.07
9	93.13	0.00	48.75	20452.89	2384.47	266.06	185.81
10	0.00	0.00	0.00	408.60	45.00	22.76	0.30
11	0.00	0.00	0.00	530.40	45.00	22.76	0.30
12	0.00	0.00	0.00	845.40	45.00	219.54	163.13
Total	**386.34**	**296.83**	**82.84**	**159447.69**	**22527.48**	-	-

The annual electric energy consumption during off peak hours was 159.45 MWh and 22.53 MWh during peak hours, totaling 181.98 MWh. The highest demand value was 266.06 kW, which occurred in March and April. In these months, the simulation showed that operations such as receiving, cleaning, drying, and dispatching occurred simultaneously. Through the present study, it was verified that 34.78 % of the electric energy was consumed in the drying sector (Table 4). This occurred because product with moisture content above 18% w.b. needs to pass through the dryer at least twice if a mixed flow dryer is used. As shown in Table 2, almost 90% of the corn and 45% of the wheat had more than 18% w.b. This high percentage of wet product is the reason more electric energy was consumed in the drying operation than in any other operation.

According to the simulation (Table 4), the drying sector had to handle 37,419.16 t of corn, 31,329.80 t of soybeans, and 8,899.01 t of wheat to dry 14,226.92 t of corn, 23,031.28 t of soybeans, and 5,042.81 t of wheat, due to product re-circulation. The drying process presented specific electrical energy consumption of 2.138, 1.109, and 1.438 kWh per tonne of corn, soybean, and wheat respectively.

CONCLUSION

A simulation toolset called *Grain Facility* was developed. It provides a suite of blocks that allow: (*i*) modeling a wide variety of grain storage facility layouts, and (*ii*) simulating dynamic behavior considering the stochastic factors that affect system performance.

To demonstrate an application of the *Grain Facility* library, a grain storage facility was modeled. It received 15,176.00 t, 5,3179.00 t, and 5,118 t of corn, soybean and wheat respectively. Simulated results show that: (i) after processing unit operations, 12,205.02 t, 47,451.81 t, and 4,356.88 t corn, soybeans and wheat were obtained respectively, (ii) the maximum amount of product stored occurred in April when the final monthly balance showed 14,409.20 t of soybean, (iii) the amount of firewood burnt in the drying operation totaled 766.02 tonnes with 27.0, 13.0, and 16.0 kg of firewood being consumed to dry each tonne of corn, soybeans, and wheat respectively, and (iv) the total facility's annual electrical energy consumption was 181.98 MWh, 34.78% of which was used in the drying sector.

By using the *Grain Facility* toolset, structured models become a more useful simulation mechanism because the toolset addresses a wide range of outputs and has significant problem-solving potential. It can be used to: (i) conduct grain storage facility feasibility analysis, (ii) define the ideal type of structures, process machines, and conveyors, (iii) detect bottlenecks, (iv) plan maintenance services, (v) optimize consumption of electricity and heat energy, (*vi*) test the impact of different operating conditions on the grain facility storage sector, and (vi) conduct simulation experiments, such as sensitivity analysis, scenario analysis, optimization, and Monte Carlo simulation.

REFERENCES

Benock, G., O. J. Loewer, T. C. Bridges, and D. H. Loewer. 1981. Grain flow restrictions in harvesting–delivery-drying systems. *Trans ASAE* 24(5): 1151-1161.

Bridges, T. C., G. M. White, I. J. Ross, and O. J. Loewer, Jr. 1982. A computer aid for management of on-farm layer drying systems. *Trans. ASAE* 25(3):

811-815.

Bridges, T. C., D. G. Colliver, G. M. White, and O. J. Loewer. 1984. A computer aid for evaluation of on-farm stir drying systems. *Trans. ASAE* 27(5): 1549-1555.

Bridges, T. C., O. J. Loewer, Jr., J. N. Walker, and D. G. Overhults. 1979. A computer model for evaluating corn harvesting, handling, drying and storage systems. *Trans. ASAE* 22(3): 618-629.

Brooker, D. B., F. W. Bakker-Arkema, and C. W. Hall. 1992. *Drying and storage of grains and oilseeds*, Chapter 10:287-291, New York: Van Nostrand Reinhold.

Imagine That, Inc. 1997. *Extend [Extend User's Manual - Version 4]*. San Jose, CA, USA

Kazmier, L. J. 1995. *Business statistics*. 3rd Edition, Schaum's Outiline Series. New York: McGraw-Hill.

Krahl, A. 2000. The extend simulation environment. *Proceedings of the 2000 Winter Simulation Conference.* Eds. J. A. Joines, R. R. Barton, K. Kang, and, P. A. Fishwick, pp 280-289. IEEE, Piscataway, NJ.

Law, A. M., and W. D. Kelton. 1991. *Simulation modeling and analysis*. 2nd Edition. New York: McGraw Hill.

Loewer, O. J., T. C. Bridges, and R. A. Bucklin. 1994. *On-farm drying and storage systems*. ASAE Publication 9, American Society of Agricultural Engineers.

Loewer, O. J., M. K. Kocher, and J. Solamanian. 1990. An expert system for determining bottlenecks in on-farm grain processing. *Applied Engineering in Agriculture* 6(11): 69-72.

Loewer, O. J., Jr., T. C. Bridges, and D. G. Overhults. 1976. Computer layout and design of grain storage facilities. *Trans. ASAE* 19(6): 1130-1137.

Maria, A. 1997. Introduction to modeling and simulation. *Proceedings of the 1997 Winter Simulation Conference Proceedings*. Eds. S. Andatóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, pp 7-13. IEEE, Piscataway, NJ.

Palisade Corporation. 2000. *@Risk: Advanced risk analysis for spreadsheets*.

Newfield, NY. USA.

Rivera, J. 1997. Modeling with Extend™. Proceedings of the 1997 Winter Simulation Conference, eds. S. Andradóttir, K. J. Healy, D. H. Healy, D. H. Withers, snd B. L. Nelson, pp 674-679. IEEE, Piscataway, NJ.

Roberts, C. A. and Y. J. Dessouky. 1998. An overview of object-oriented simulation. *Simulation* 70(6): 359-368. 1998 Simulation Councils, Inc.

Thompson, T. L., R. M. Peart, and G. H. Foster. 1968. Mathematical simulation of corn drying - A new model. *Trans. ASAE* 11(4): 582-586.

Weber, E. A. 2001. *Armazenagem agrícola [Farm Storing].*Guaíba, RS. Brasil: Livraria e Editora Agropecuária