3B2v8.06a/w (Dec 5 2003).51c +model | EAAI : 970 | Prod.Type:FTP pp.1–17(col.fig.:NIL) | ED:RajeshGupta PAGN:savitha SCAN:

ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Engineering Applications of Artificial Intelligence ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Automatic generation of explanations: AGE

Silvia B. González-Brambila[a,*], Eduardo F. Morales[b]

[a]Systems Department, UAM-Azcapotzalco, 02200, D.F. MÉXICO, México
[b]Computer Science, INAOE, 72840, Sta. María Tonantzintla, MÉXICO, Puebla

Received 10 June 2005; received in revised form 11 April 2006; accepted 29 May 2006

Abstract

Explaining how engineering devices work is important to students, engineers, and operators. In general, machine generated explanations have been produced from a particular perspective. This paper introduces a system called automatic generation of explanations (AGE) capable of generating causal, behavioral, and functional explanations of physical devices in natural language. AGE explanations can involve different user selected state variables at different abstraction levels. AGE uses a library of engineering components as building blocks. Each component is associated with a qualitative model, information about the meaning of state variables and their possible values, information about substances, and information about the different functions each component can perform. AGE uses: (i) a compositional modeling approach to construct large qualitative models, (ii) causal analysis to build a causal dependency graph, (iii) a novel qualitative simulation approach to efficiently obtain the system's behavior on large systems, and (iv) decomposition analysis to automatically divide large devices into smaller subsystems. AGE effectiveness is demonstrated with different devices that range from a simple water tank to an industrial chemical plant.

Keywords: Qualitative simulation; Causal analysis; Explanations; Qualitative differential equation; Landmark; Flowsheet; Functional analysis

1. Introduction

Communicating knowledge, in verbal or written form, is an important human learning activity. In engineering, explaining how a particular device works is relevant to engineering students, designers and operators of industrial plants. These explanations, however, are normally given from a particular point of view and without considering the user's particular needs. Machine generated explanations of physical devices normally considered a particular perspective (e.g., functional identification (Kitamura et al., 2002)). Explanations related to a particular device can be given from different perspectives depending on different needs. An engineer may be interested in knowing the causal dependencies between different state variables. She may be interested in observing how the state variables evolve over time, or what is the main function of a particular device. Her interests may focused on particular state variables and/or particular subsystems. All these explanations are important and provide complementary information to a user. This paper describes a system called automatic generation of explanations (AGE), which can produce explanations of engineering devices in natural language considering different perspectives. In particular, AGE produces causal, behavioral and functional explanations, considering user selected state variables and subsystems.

The goal of AGE is to create understandability through the generation of natural language descriptions produced by several inferences processes like causal order, qualitative simulation and subsystem reduction.

This paper is organized as follows. Section 2 describes the general architecture of AGE and how it produces its different explanations. An evaluation of AGE in terms of applicability and usability is given in Section 3. Section 4 reviews related work and Section 5 provides conclusions and future research directions.

*Corresponding author. Tel.: 525553189532; fax: 52553944534.
E-mail addresses: sgb@correo.azc.uam.mx (S.B. González-Brambila), emorales@inaoep.mx (E.F. Morales).

0952-1976/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.engappai.2006.05.011

2. AGE

Physical devices are specified in AGE by joining individual engineering components, such as pumps, valves, tanks, tubes, stoppers, reactor, etc., taken from a library of components through a graphical interface or alternatively by selecting a previously constructed device. Joining components was introduced by Gruber and Oisen (1994). In AGE, each component of this library is associated with a qualitative model as in QSIM (Kuipers, 1994). We adopted qualitative models because they provide an adequate abstraction level from which useful explanations in natural language can be easily produced, and they allow predictions about the behavior of the system in the absence of exact quantitative information.

A qualitative differential equation (QDE) model is an abstraction of an ordinary differential equation, consisting of a set of real-valued variables and functional, algebraic and differential constraints among them, where the values of variables are described in terms of their ordinal relations with a finite set of symbolic landmark values, rather than in terms of real numbers. A quantity space is a finite, totally ordered set of symbolic *landmark values* representing qualitatively important values in the real number line (Kuipers, 1994).

The complete specification of a physical component in AGE, requires, besides a qualitative model, the semantic meaning of each state variable and all of its landmark values, as well as its input/output variables in order to connect it with another component. For instance, Fig. 1 shows semantic information (in Spanish) associated with a tank filled with water. Each component is also associated with a meaningful name to the user and the name of the substance that it is carrying. In case of chemical reactions within the component, it is the user's responsibility to specify the products.

AGE follows a compositional modeling process (e.g., see Falkenhainer and Forbus, 1991) to construct a global qualitative model that takes into account conservation of mass and energy (e.g., the pressure is assumed to be constant between components and all the input and output flow variables of a particular component must sum zero). AGE also identifies the exogenous variables.

AGE's architecture, once a global qualitative model has been constructed, is shown in Fig. 2. Given a qualitative model of a particular device, AGE: (i) generates a global flow sheet that is used for functional explanations, (ii) obtains causal dependencies from the qualitative model to produce causal explanations, (iii) simulates the qualitative model to produce behavioral explanations, and (iv) uses this simulation with functional analysis to produce functional explanations. The following subsections explain each of these steps in more detail.

2.1. Causal explanations

An intuitive explanation of a device can be given in terms of causal dependencies of state variables. Given a set of

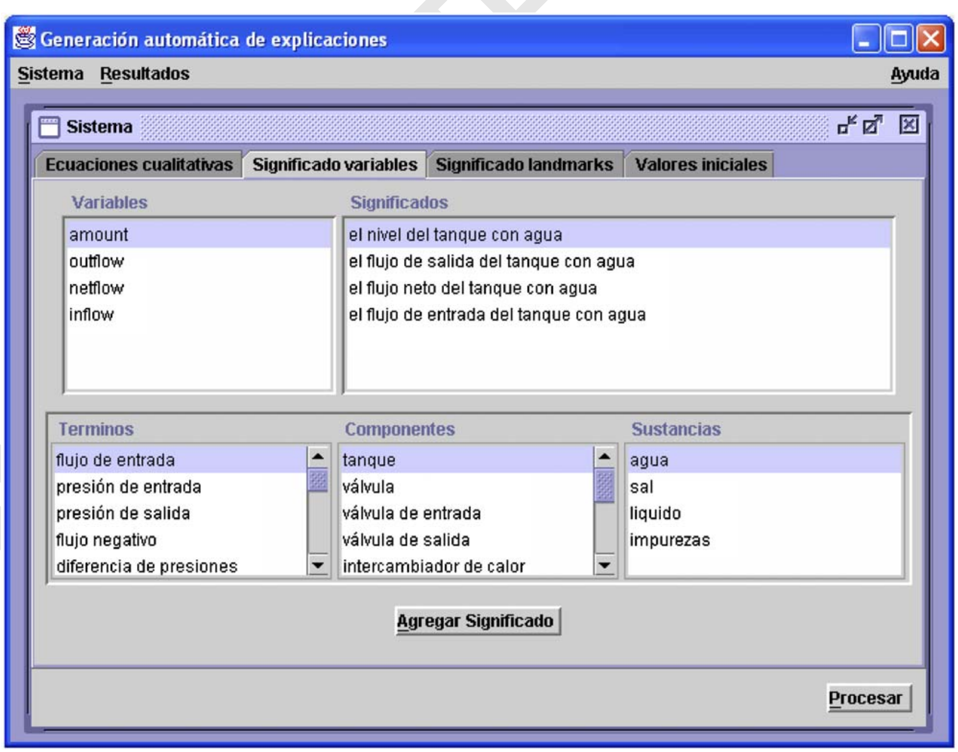


Fig. 1. Semantic information associated with each state variable. Each variable (e.g., amount highlighted in the upper half) has information about its meaning (*the amount of water in the tank*), and information about the substance and related component (e.g., *input flow* highlighted in the lower half, is associated with a particular *tank* and with *water*).

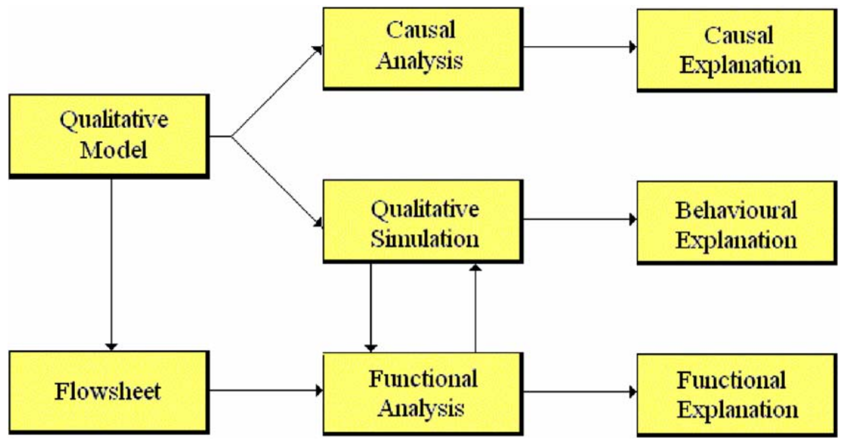


Fig. 2. AGE's architecture.

exogenous variables, causal knowledge can be derived from a set of equations using Iwasaky and Simon's algorithm (Iwasaki and Simon, 1993). The general idea is to build the minimal self-contained system (minimum set of independent N equations with N variables) and link it to the next self-contained system until all the equations are considered (see Iwasaki and Simon, 1993 for more details).

In a causal graph there is a node for each state variable and a directed link between variable X and Y ($X \rightarrow Y$) if the values of Y depends on the values of X.

AGE uses a modified version of this algorithm for qualitative models, which in AGE can be over determined (i.e., with redundant equations) due to the compositional modeling process, so the causal order is not unique. Starting with exogenous variables, links are collocated in accordance to the syntax of the equation. So, this depends on the number of parameters of each restriction.

Consider, for example, a valve represented by the following QDE using prefix notation, where M+(x,y) means that x increases monotonically with respect to y. This is over determined because there are six variables and 7 QDEs:

(1) constant (k)
(2) constant (qIn)
(3) M+ (qIn qOut)
(4) *(qIn k dp)
(5) M+ (qIn pIn)
(6) M+ (qOut pOut)
(7) + (dp pOut pIn)

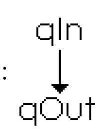
The meaning of the variables is shown in Table 1:

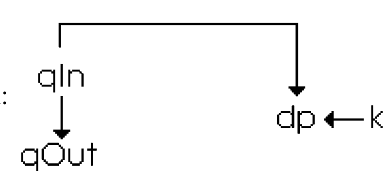
Table 1
Meaning of the variables used to represent a valve

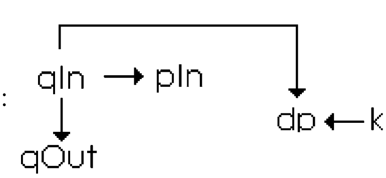
Variable	Meaning
k	Valve constant
qIn	Inflow
qOut	Outflow
pIn	In pressure
pOut	Out pressure
dp	Pressure difference

The semantic information associated with each state variable depends on its meaning, the substance and the related component.

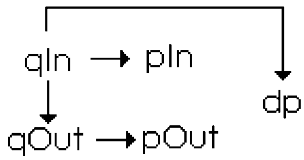
The steps followed by the causal order algorithm are in this case:

1. *DeterminedVariables*←k, from Eq. (1)
   Causal graph: k
2. DeterminedVariables←qIn∪DeterminedVariables, from Eq. (2)
   Causal graph: qIn
3. DeterminedVariables←qOut∪DeterminedVariables, from Eq. (3)
   Causal graph: 

4. DeterminedVariables←dp∪DeterminedVariables, from Eq. (4)
   Causal graph: 

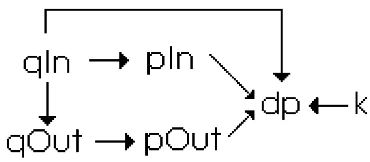
5. DeterminedVariables←pIn∪DeterminedVariables, from Eq. (5)
   Causal graph:: 

6. DeterminedVariables←pOut ∪ DeterminedVariables, from Eq. (6)

Causal graph: 

At this stage all the system variables are included in the graph, but Eq. (7) has not been used yet. The last step:

7. DeterminedVariables ← DeterminedVariables, from equation (7)

Causal graph: 

This type of graph is used to produce causal explanations, as we will see later. One of the advantages is that the same graph can be used for different languages.

Let us consider a tank and two valves shown in Fig. 3. A possible qualitative model of this system is shown in Table 2, its causal graph is shown in Fig. 4, and its causal explanation is given in Fig. 5. AGE produces syntactically correct textual explanations (in Spanish) considering punctuation, gender and number agreement, and eliminations of text (called reductions) to avoid unnecessary repetitions. For example, if a variable depends on another variable of the same component and substance, the component and substance are left implicit and are not mentioned again during the explanation associated with the component. The user can also select particular variables to consider in the explanations.

To produce causal explanations, AGE traverses the causal graph using breadth-first search, considering exogenous variables first and taking care of possible cycles.

Explanations are produced in reverse order, where the last node (which depends on the rest of the variables) is used first in the causal explanations. The explanation continues until reaching an exogenous variable. To produce causal explanations in natural language, the semantic meaning of each state variable, component and substance is consulted and used to fill-in text templates.

When the user selects a subset of variables, AGE constructs a causal graph only with these variables and with their neighbor variables represented in the causal graph. For example, if the user selects only input and output variables of the two valves and a tank system, the causal graph consider for the explanation and the explanation are shown in Fig. 6.

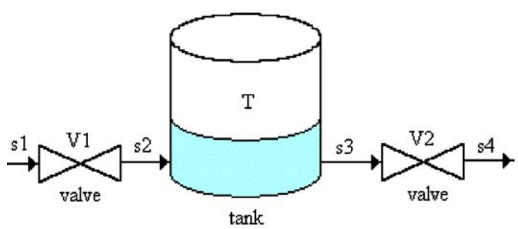


Fig. 3. Two valves and a tank.

Table 2
Qualitative model after compositional modeling of a tank and two valves

Tank system
minus (V1-MinusQ T-Inflow)
minus (T-Outflow V2-Q)
add (V1-dp T-Pin V1-Pin)
constant (V1-K ∣ k)
M- (V1-Q V1-MinusQ ∣ 0 0 q –q)
mult (V1-Q V1-K V1-dp ∣ 0 k 0)
M+ (V1-Q V1-Pin ∣ 0 0 q p)
M- (V1-MinusQ T-Pin ∣ 0 0 -q p)
M- (T-Amount T-Outflow ∣ 0 0 full –q)
M- (T-Outflow T-Pout ∣ 0 0 -q p)
add (T-Netflow T-Outflow T-Inflow)
deriv (T-Amount T-Netflow)
M+ (T-Inflow T-Pin ∣ 0 0 q p)
add (V2-dp V2-Pout T-Pout)
constant (V2-K ∣ k)
M- (V2-Q V2-MinusQ ∣ 0 0 q –q)
mult (V2-Q V2-K V2-dp ∣ 0 k 0)
M+ (V2-Q T-Pout ∣ 0 0 q p)
M- (V2-MinusQ V2-Pout ∣ 0 0 -q p)

Where V1 = valve1, T = tank, V2 = valve2, Q = flow, K = constant, D = delta (diff.), and P = pressure, where q (Ar_1, Ar_2, ! ∣ v_1, v_2, !), q = qualitative constraint (ejem. add, minus, constant, etc), Ar_i are qualitative variables, v_i are landmark values.

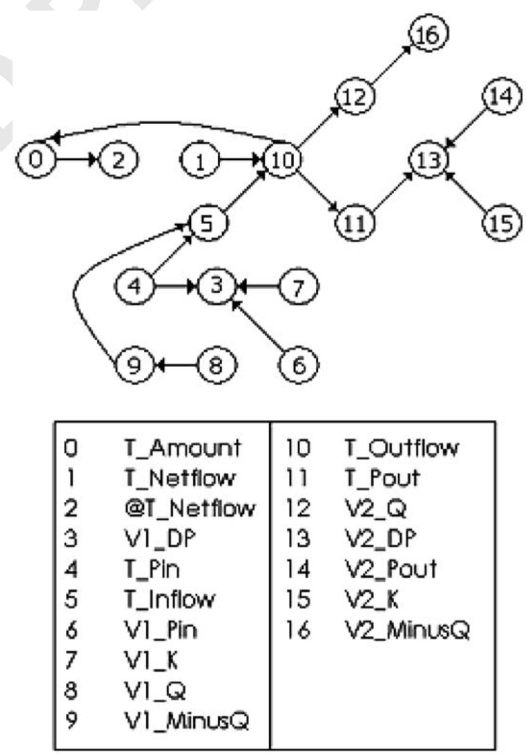
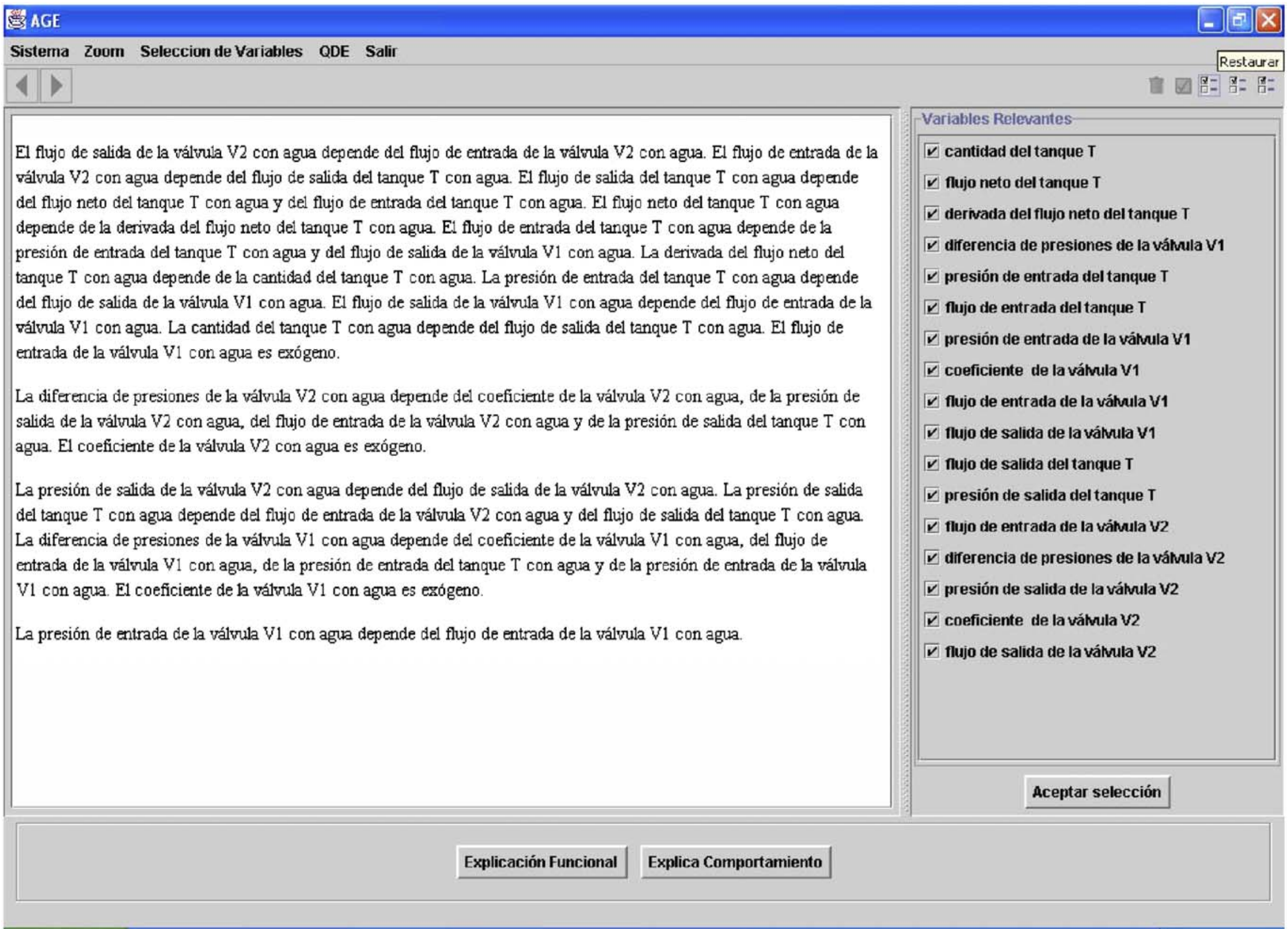


Fig. 4. Causal graph without cycles of the qualitative model of a tank and two valves where @ means derivate.



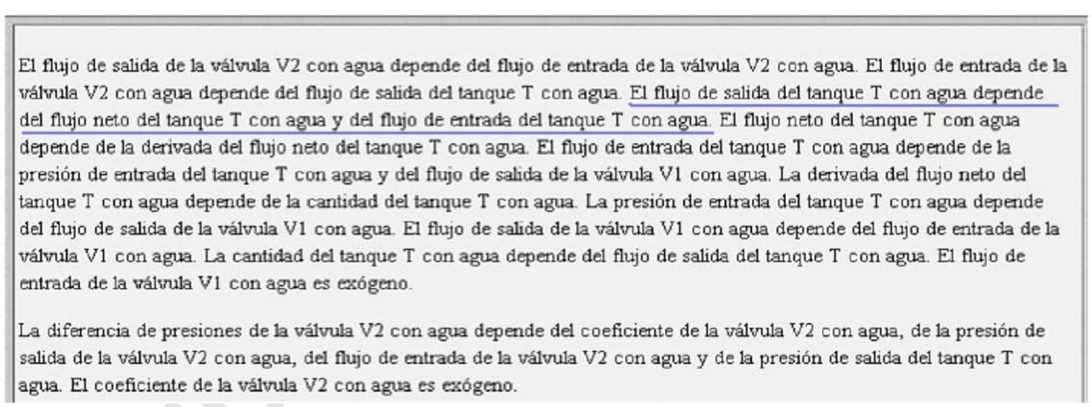


Fig. 5. Causal explanation for a tank with two valves. The first sentence says roughly …*The output flow of the tank T depends on the netflow of tank T with water and the inflow of tank T with water*… (this corresponds to nodes 1, 5 and 10 of Fig. 4) (a) Causal graph and (b) Causal explanation.

2.2. Behavioral explanations

In order to produce behavioral explanations, for all possible starting states, AGE produces a behavioral graph (a graph where each node represent a particular qualitative state and links represent time sequences) using a modification of QSIM (Kuipers, 1994). AGE traverses the behavioral graphs to produce behavioral explanations using information about each state variable, component and substance (e.g., see Fig. 1), and the semantic meaning of all the landmarks associated to each state variable. AGE uses text templates and syntactic considerations to produce meaningful and syntactically correct explanations.

Each time the behavioral graph branches, AGE creates hypertext links for each branch to facilitate the understandability of the possible qualitative behaviors. For instance, Fig. 7 shows tree possible qualitative behaviors of a tank, while Fig. 8 shows the explanation produced by AGE, if the user selects the middle link.

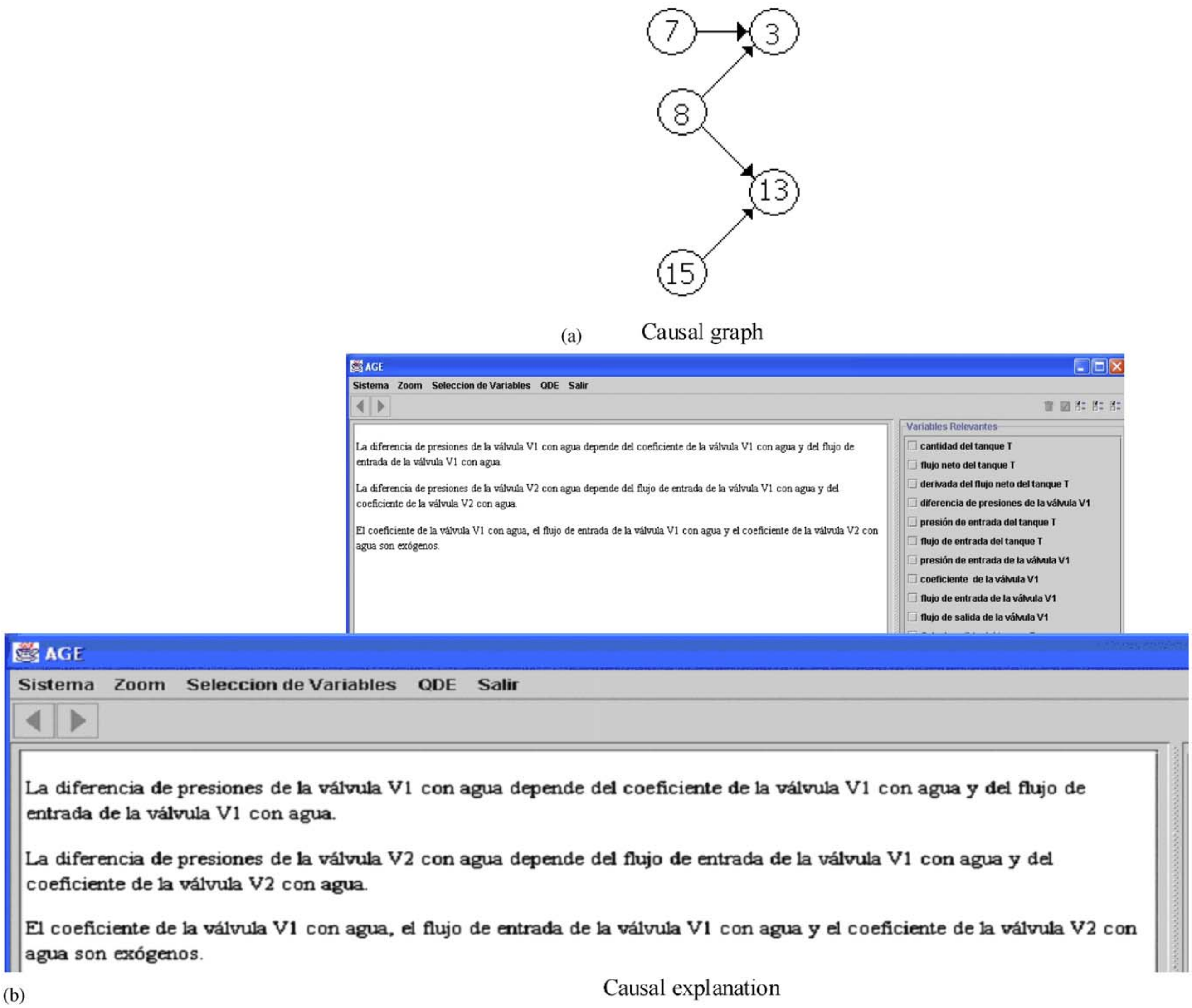


Fig. 6. Causal explanation produced when only input and output variables are selected by the user. Input variables are: V1-K (7), V1-Q (8) y V2-K(15) and output variables are: V1-dp (3) and V2-dp (13).

Behavioral explanations are simplified when a state variable follows an increasing (decreasing) behavior through several qualitative states and across several landmark values. For example, consider the consecutive landmark values *LandMark1*, *LandMark2*, …, *LandMarkn* and the following sequence: "*Var1* in *LandMark1* and increasing, *Var1* between *LandMark1* and *LandMark2* and increasing, *Var1* in *LandMark2* and increasing…, *Var1* in *LandMarkn* and constant". This sequence is reduced to: "*Var1* increases from *LandMark1* to *LandMarkn*". The user can also produce new explanations by selecting particular variables.

Qualitative simulation is very important in AGE, because functional and behavioral explanations are generated from it. AGE produces a behavioral graph (a graph where each node represent a particular qualitative state and links representing time sequences) using a re-implementation of QSIM (Kuipers, 1994). QSIM can be very inefficient for large systems. For instance, Catino in (1993) simulated in 12 h a nitric acid plant with 217 variables and 287 constraints in a 224 MB Sun SparcStation ELC using QSIM. In our re-implementation of QSIM we were not able to simulate a chemical plant with 88 components after one day of CPU time (Intel Pentium III 993 MHz, 256 MB). This paper introduces an extension to QSIM which divides each system into smaller subsystems considering design principles of process engineering. Individual components are simulated qualitatively from which their behavioral graph are produced. The algorithm joins these graphs and continues until a complete simulation is obtained, it is shown that our approach achieves substantial reductions in computational time allowing to simulate industrial plants in a few minutes.

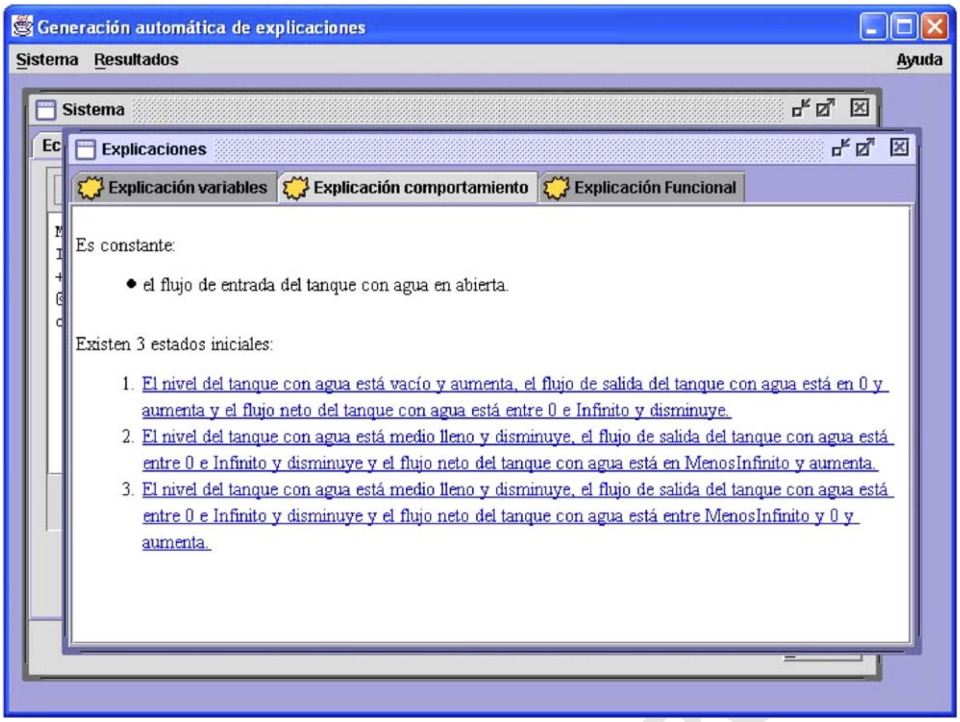


Fig. 7. Three possible behavioral explanations of a tank. The text says that there are tree possible initial states, the first one says roughly: *The water level of the tank is empty and increasing, the outflow of the tank is 0 and increasing and the net flow of the tank is between 0 and infinite and decreasing.*

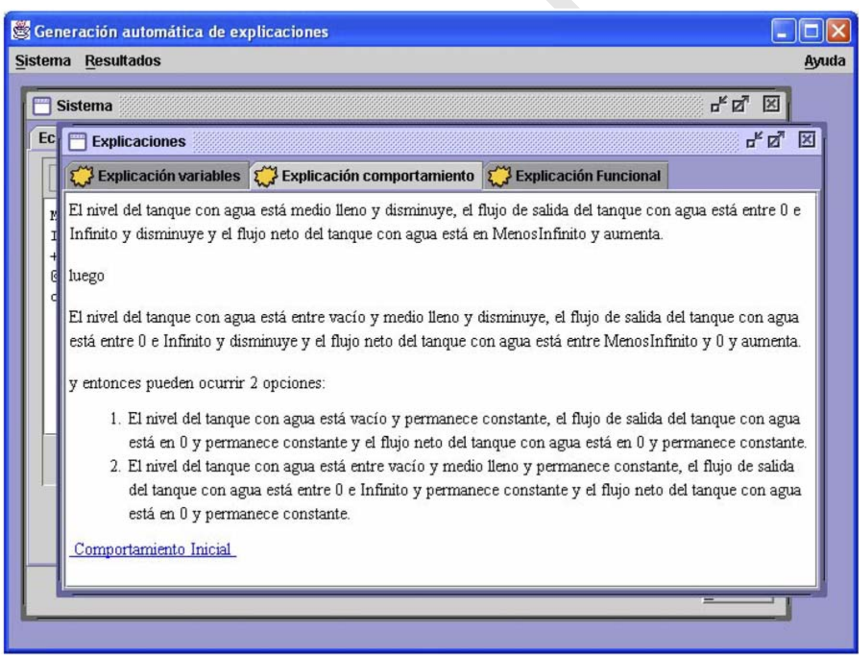


Fig. 8. Explanation of one possible qualitative behavior of the tank. The text roughly says: The water level of the tank is half-full and decreasing, the output flow is between 0 and infinite and decreasing …then the water level of the tank is between empty and half-full and decreasing …and then there are 2 possible options: …

2.2.1. QSIM

QSIM is an approach to qualitative simulation that uses QDE to represent a system. QDE are relaxed versions of ordinary differential equations (Kuipers, 1994). QSIM predicts the possible behavior set of a QDE. A QDE model is qualitative in two senses. First, the values of variables are described in terms of their ordinal relations with a finite set of symbolic landmark values. Second, functional relations may be described as monotonic functions (Kuipers, 2001). Landmark values are the "natural joints" that break a continuous set of values into qualitatively distinct regions. A landmark value is a symbolic name for a particular real number, whose numerical value may or not be known. It serves as a precise boundary for a qualitative region.

QSIM starts with a QDE and a qualitative description of an initial state. Given a qualitative description of a state, it predicts the possible qualitative state descriptions that can be direct successors of the current state description. Repeating this process produces a graph of qualitative descriptions, in which the paths starting from the root are the possible qualitative behaviors. The resulting behavior graph however can be huge.

The main step in the QSIM algorithm is to generate all the successor states given a state. The successor generation algorithm performs the following steps (see Kuipers, 1994):

1. Domain restriction
2. Node consistency
3. Arc consistency
4. Exhaustive search
5. Filtering

To guarantee that all possible behaviors are predicted, it is required that all possible qualitative value transitions are predicted, and that the combinations of qualitative values are only deleted when they are inconsistent. The exhaustive nature of the QSIM simulation can produce excessive running times.

When a qualitative model of a component is defined, it is very important to analyze the possible landmarks of each variable, the initial conditions and the constraints with the corresponding values because the execution time depends on all of these factors.

2.2.2. *Subsystem reduction*

For the subsystem reduction process, principles from classical design in process engineering (e.g., Beltrán et al., 1997; Douglas, 1988) were considered, where the component's system are collocated in accordance to their type. This means that in a new design the first components that are considered are the reactors, then separators, energy transfers units, material management units and lastly, the rest of the equipment.

In our case, units are grouped together using the priority list, shown in Table 3. For example, it is common to mix 2 or more substances (mixer), heat the product (heater) and finally introduced the product into a reactor. These three units (mixer, heater and reactor) can be merged in one subsystem whose purpose is to react.

Two units, *A* and *B*, can be merged into a subsystem *A–B* if *A* is adjacent to *B*, *A* has a priority equal or smaller than *B*, and *A* is topological smaller than *B*. In the topological sort each node is associated with a vertex and there is a directed edge from node *x* to node *y* if *y* cannot start until *x* has finished.

A large system can go through several grouping processes, so this is an iterative process, that ends when there is only one system. After the first unit is selected the system tries to group it with its neighbors. A unit is considered first if it has more external substances, lower priority type and is first in the topological sort of all the system.

Consider the flow sheet of the hydrodealkylation of toluene shown in Fig. 9. Grouping the units result in the systems shown in Figs. 10 and 11. Fig. 10 is the first iteration of the algorithm, the *reactor system* groups the *compressor*, *pump*, *mixer* and the *reactor*; the *separation system* adjacent to the *reactor system* contains the *flash* and the *separator*, the *separation system 1* groups *valve1* and *separator1* and *separation system 2* groups the rest of the units. Note that the cycles in the reduced subsystems are maintained.

Table 4 shows the main steps to reduce subsystems. It selects the initial node without considering the substances. This initial node is inserted in an empty new flow diagram. Then the algorithm tries to reduce the number of units with the nearest neighbors; this depends on the functional priority of each unit. When two or more units are grouped together in one subsystem, they are inserted into a list in order to save this information for later. These grouped units are inserted like nodes in the graph that represents a new flow diagram. Finally the arcs of the new flow diagram are inserted considering the new subsystems and the substances.

Figure 11 is obtained from Fig. 10; here Reactor system, Separation system and Separation system 2, from Fig. 10, are grouped into one Reactor system. With this reduction there are two subsystems: reactor and separation, and nine substances.

In Fig. 12 all subsystems are grouped into one, where only input and products substances are considered, this is the last iteration.

Table 3
Priority of unit type

Priority	Unit type	Examples
1	Reactor	All types of reactors
2	Separator	Filters, evaporators, centrifuges
3	Energy transfer	Heaters, coolers
4	Material management	Pumps, mixers, compressors, turbines
5	Storage and control	Tanks, valves

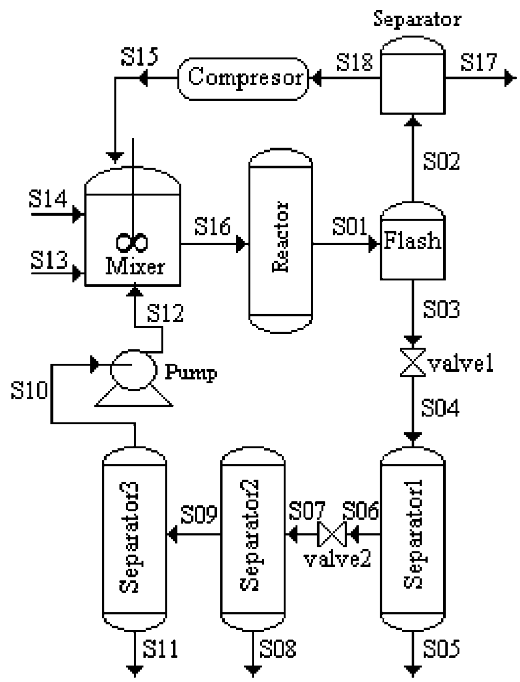


Fig. 9. Hydrodealkylation of toluene.

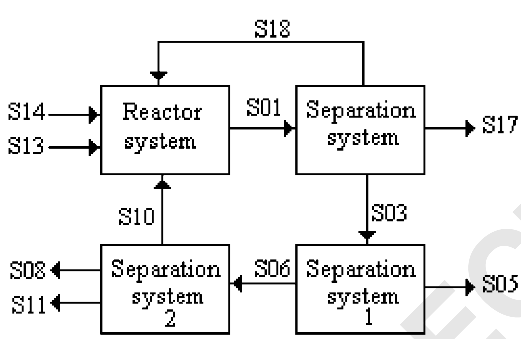


Fig. 10. Hydrodealkylation of toluene first iteration of the subsystems reduction process.

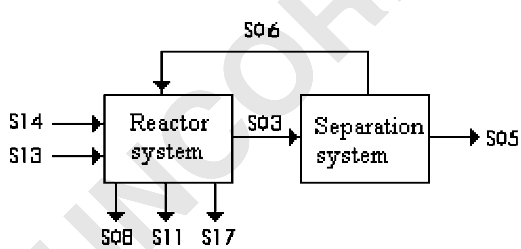


Fig. 11. Subsystems of the Hydrodealkylation of toluene.

2.2.3. Simulation by components

Our algorithm simulate individual behaviors of each component in the system using QSIM. This process produces behavioral graphs for each component. Individual behavior graphs are grouped in subsystems, using the subsystem reduction algorithm.

Table 4
Subsystem reduction algorithm

```
FlowDiagram SubsystemReduction ( )
{
  initialNode←select begin unit
    //Create a new empty diagram called nD
  FlowDiagram nD←new FlowDiagram ( )
  insert initialNode in nD
  reduceUnits (initialNode, nD)
    //insert links in accordance to
    //the previous flow diagram
  nD.putLinks (this)
    //insert substances in accordance to
    //the previous flow diagram
  nD.createSubstances (this)
}
```

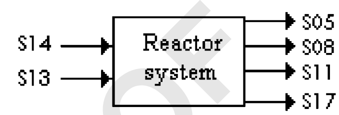


Fig. 12. Last iteration of the subsystem reduction of the Hydrodealkylation of toluene.

The main idea is to divide the system in subsystems at different abstractions levels, use QSIM to simulate each individual component and obtain their respective behavior graph considering different abstractions levels. The behavior graphs are grouped by the connection nodes in the subsystems and only when all their behavior values are equal. Even though it is possible to generate more states than necessary in the component level, they will be eliminated during the union process and significant reductions in execution time are obtained.

To group two different behavior graph nodes, both nodes must correspond in time and the union qualitative values must be equal (*terminalIn* or *terminalOut*).

For example, suppose we have a unit A with a behavior graph $g1$ with an initial node "a" with set values $\{v\_a\}$, where $\{v\_a\}$ corresponds to all the qualitative variable values of unit A at time $t0$. Now suppose we have a unit B with behavior graph $g2$ and an initial node "b" with set values $\{v\_b\}$. In addition, consider A to be before unit B in the topological sort of the flow sheet. Since a and b are initial states they both occur at time $t0$. If the *terminalOut* qualitative values in {v_a} are equal to *terminalIn* qualitative values in {v_b}, then they can be merge into one state. This new state contains all the values in {v_a} and all the values in {v_b}, except those in the intersection of *terminalIn* in A and *terminalOut* in B, that are considered only once. The remaining nodes are merged in a similar form (see Fig. 13).

A final node is considered quiescent, if the variable values are the same in the following transition, these nodes are called durables. In the case of merging, a behavior graph with only one state (with a value set $\{a\}$) with another graph with several nodes, the single node needs to

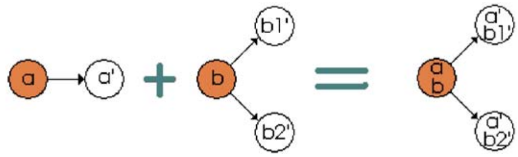


Fig. 13. Example of joining two-behavior graphs.

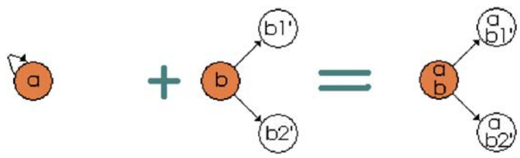


Fig. 14. Considering a perdurable node.

Table 5
Mixer behavior

(a) Behavior graph	
State	Adyacents
M0	—
(b) Values of the state	
Variable	State M0
M-amount	half,θ
M-outflow	q, θ
M-netflow	0, θ
M-inflow	q, θ
M-Qin1	q, θ
M-Qin2	q, θ

be mapped with all the nodes of the other behavioral graph (see Fig. 14). So the mapping process is in general *1* to *N*, because one node can be consider more that one.

2.2.4. Example

Consider a system with a mixer and a reactor, called *MR*. Suppose that the connecting variables are only the outflow of the mixer (*M-outflow*) and the inflow of the reactor (*R-Fin*). Suppose that the input flow of the mixer is constant in order to reduce its possible behaviors. The behavior graph of each component is presented in Tables 5 and 6, respectively. Table 6 shows part of the behavioral graph represented in list form, some of the initial states are *R*0, *R*1, *R*2, *R*3, *R*4, *R*5, *R*6. The QSIM simulation produced 32 states in total.

Initially consider the state *M*0, the only mixer initial state, and the reactor initial state *R*6. With these two states we construct a new one (*MR*6) of the behavior graph of the *MR* system. This is possible because *M-outflow* and *R-Fin* have the same value (*q*, *θ*). First column of Table 7 shows the values of this state.

Next we consider state *R*17, because it is adjacent to *R*6. With *M*0 and *R*17 another new node of the behavior graph is constructed. In this case *MR*6 and *MR*17 must be adjacent, so the behavior graph of the system is constructed with these nodes linked together (see Fig. 15(a)). In the construction of this new state, *M*0 is considered durable.

The construction process of the behavior graph continues with the adjacent nodes of *M*0 and *R*17, which are *M*0 (durable) and *R*13, respectively. These new states are merged and a new is created state *MR*13, adjacent to *MR*17 (see Fig. 15(b)).

This process continues until all nodes are visited, when the process finish the graph contains three nodes and two links.

The algorithm in the worst case is $O(n^2)$, without considering the QSIM simulation of individual components, because it uses a depth first search in which the nodes of the second graph can be visited more than once.

We have observed in all of our experiments that our merging procedure produces only qualitatively consistent behavioral graphs, and as part of our future work, we are working on a formal proof of this.

By joining individual behavioral graphs of single components, we are able to substantially reduce the computational time required by QSIM. Section 3 describes in more detail some of the reductions in time achieved with this approach.

2.3. Functional explanations

Each component in AGE is associated with its main function or purpose and a list of secondary functions that may apply under particular circumstances. For instance, the main purpose of a mixer is to mix substances, however, if there is a temperature difference between input substances that are equal, it can be used as a heat exchanger, because the output substance has only one temperature. To decide which particular function a component is performing, AGE uses information from the behavioral graphs. Although a device may be associated with a particular function, this will not be reported by AGE if it does not comply with its expected behavior (i.e., there is a direct correspondence between behavioral graphs and associated functionality). Also, it is possible to associate one component with several functions.

This information is provided by the user. AGE produces two types of functional explanations. What we refer to *black box* functional explanations, are given in terms of which substances are received and produced by particular components and a. This information is given by the user for each system (see Fig. 1). A more detailed functional explanations, which is called *screen or grille box*, which considers behavioral information, that is information between a particular function of a device and its expected behavior. Fig. 16 shows an instance of the latter where again syntactic considerations and reductions are employed. It is produced from an acyclic process (Felder, 2000), that involves a mixer, a pump, a heater, a distillation column, and a condenser (see Fig. 17). AGE recognizes that there is a heat exchange in the mixer, due to the behavioral graph. It also simplifies the textual explanation by avoiding unnecessary references. For instance, "*Este flujo alimenta a la bomba B*1*, se calienta, se destila y se*

Table 6
Segment of the reactor behavior

(a) *Segment of the behavior graph*

State	Adjacents
R0	
R1	—
R5	—
R6	17, 18
R13	19, 20
R17	13, 14, 8, 9, 10
R19	7, 12, 10

(b) *Values of some states*

Variable	State R0	State R6	State R13	State R17
R-dif	0, ↑	0,↑	dif, θ	<0, dif>,↑
R-Ca	0, ↑	0, ↑	c, θ	<0, c>,↑
R-Fin	q, θ	q, θ	q, θ	q, θ
R-Fout	0, ↑	0, ↑	q, θ	<0, q>,↑
R-Cb	c, θ	c, θ	c, θ	c, θ
R-k	k, θ	k, θ	k, θ	k, θ
R-kCa	0, ↑	0, ↑	kc, θ	<0, kc>,↑
R-MkCb	0, ↓	0, ↓	-kc, θ	<-kc, 0>, ↓
R-D	0, ↑	0, θ	0, θ	0, θ

Table 7
Some states of the MR system

Variables	MR6	MR17	MR13
M-amount	some,θ	some,θ	some,θ
M-outflow	q, θ	q, θ	q, θ
M-netflow	0, θ	0, θ	0, θ
M-inflow	q, θ	q, θ	q, θ
M-Qin1	q, θ	q, θ	q, θ
M-Qin2	q, θ	q, θ	q, θ
R-dif	0,↑	<0, dif>,↑	dif, θ
R-Ca	0, ↑	<0, c>,↑	c, θ
R-Fin	q, θ	q, θ	q, θ
R-Fout	0, ↑	<0, q>,↑	q, θ
R-Cb	c, θ	c, θ	c, θ
R-k	k, θ	k, θ	k, θ
R-kCa	0, ↑	<0, kc>,↑	kc, θ
R-MkCb	0, ↓	<-kc, 0>, ↓	-kc, θ
R-D	0, θ	0, θ	0, θ

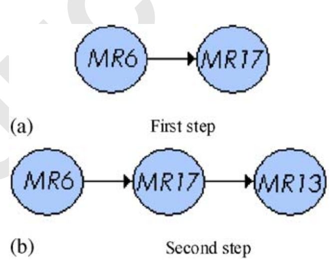


Fig. 15. Constructing behavior graph of *MR* system.

*producen s*7 *y s*8" (This flow is given to pump B1, it is heated, distilled, and s7 and s8 are produced), uses "This flow" in reference to the previously mentioned flow, and the flow is not longer mentioned while it goes through the heater and distiller, until new flows are produced. AGE also recognizes the functionality of the heater and the distiller. Again, AGE use text templates with additional syntactic rules to produce more natural outputs.

In order to understand how a large device works, it is normally required to divide it into subsystems. AGE automatically divides a large system into subsystems using information of the type associated with each individual component using traditional engineering process design priorities (see Table 4).

Individual components are grouped considering their priority, where lower priority components are grouped into higher priority components. AGE keeps track of the different components involved in each subsystem and is able to produce functional explanations (following hypertext links) at different abstraction levels.

This algorithm is useful for acyclic and cyclic process, like those shown in Figs. 17 and 18, respectively; and for small and large systems, like those shown in Figs. 19–21.

AGE uses eight templates to generate functional explanations along with several grammatical rules to produce syntactically correct sentences.

Functional explanations can be produced at any abstraction level with links to less abstracted levels. The user can select particular subsystems and or substances involved in the explanations.

3. Evaluation

In this section, we evaluate: (i) the subsystem reduction approach, (ii) the applicability of AGE to different

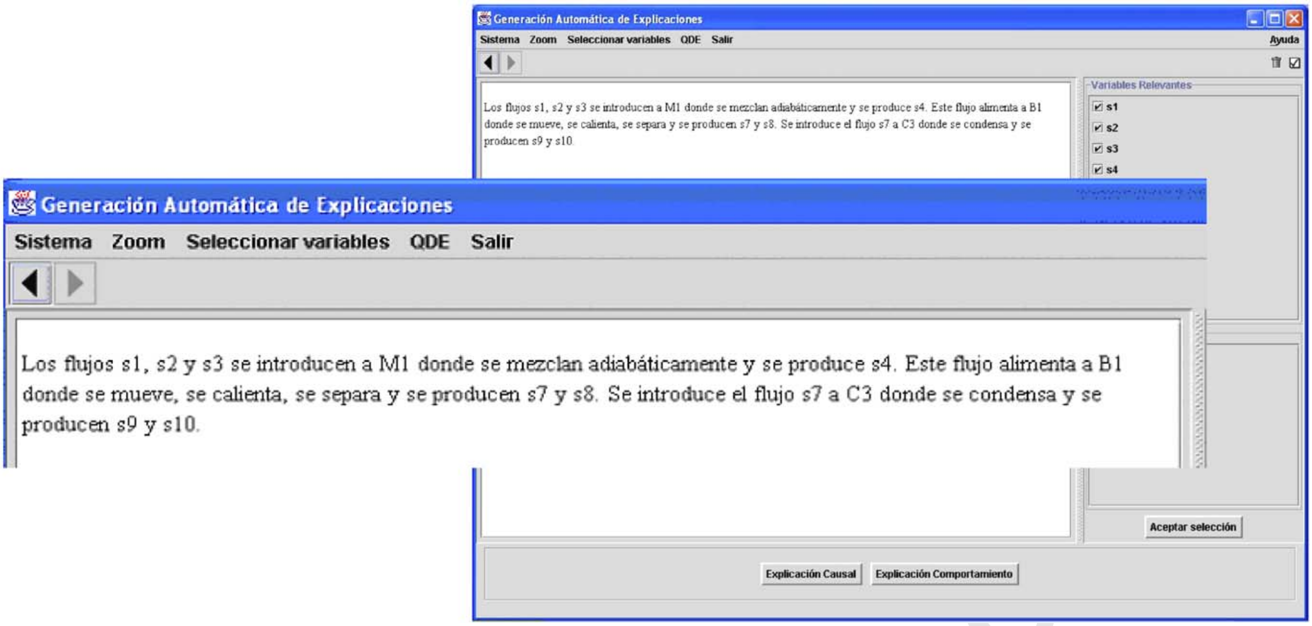


Fig. 16. Functional explanation considering behaviors (screen box). The first sentence says: The flows s1 at low temperature, s2 at high temperature and s3 at medium temperature are introduced into the mixer M1 where there is a heat exchange and s4 is produced at medium temperature.

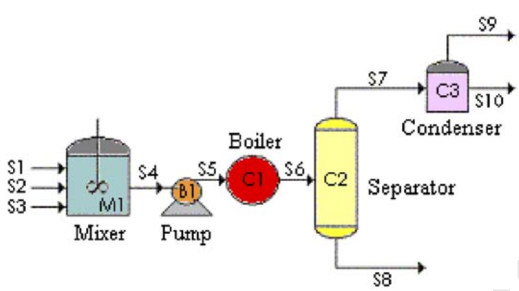


Fig. 17. A small acyclic device.

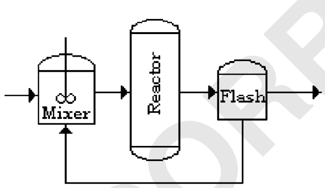


Fig. 18. MRF system, a cyclic physical device with three components.

engineering domains, and (iii) the utility of AGE to engineers.

3.1. Evaluation of the subsystem reduction approach

The subsystem reduction approach can significantly reduce the processing time and it is possible to store behavioral graphs of individual components and re-use them in other systems, reducing even more the processing time.

AGE has been tested on a wide variety of engineering systems ranging from single components to industrial plants. Figs. 18–21 show some of the processes that have been used to evaluate AGE performance. In all of them, we were able to obtain significant time reductions using our subsystem reduction approach.

For example, the Empress plant (Himmelblau and Bischoff, 1992), shown in Fig. 21, has 132 flows, 88 units and 638 variables in its qualitative representation. The average execution time of AGE considering the system reduction is 404,343.4 ms (6.74 min) using an Intel Pentium III at 993 MHz with 256 MB. The average time to simulate individual components is 129,868.7 ms (2.16 min). This is a huge reduction in processing time, considering that we were not able to simulate this plant with our re-implementation of QSIM, without subsystem reduction, after 1 day of CPU work. This is very reasonable time considering the size of the plant.

3.2. Evaluation of AGE applicability

Using AGE library of components and manually constructed systems, AGE was able to produce causal, behavioral and functional explanations for a wide variety of systems ranging from individual components to industrial chemical plants. Some of these systems involve cyclic flows, different substances, and a wide range of equipment (see González-Brambila and Morales, 2003a).

AGE's applicability can be easily extended by introducing new components into its library. The user needs to define a qualitative model and the associated information to produce different explanations.

To validate the explanations produced by AGE and assess its utility and understandability, a group of student (23) and chemical engineers (9) from the Universidad Autónoma Metropolitana, in México City, was selected.

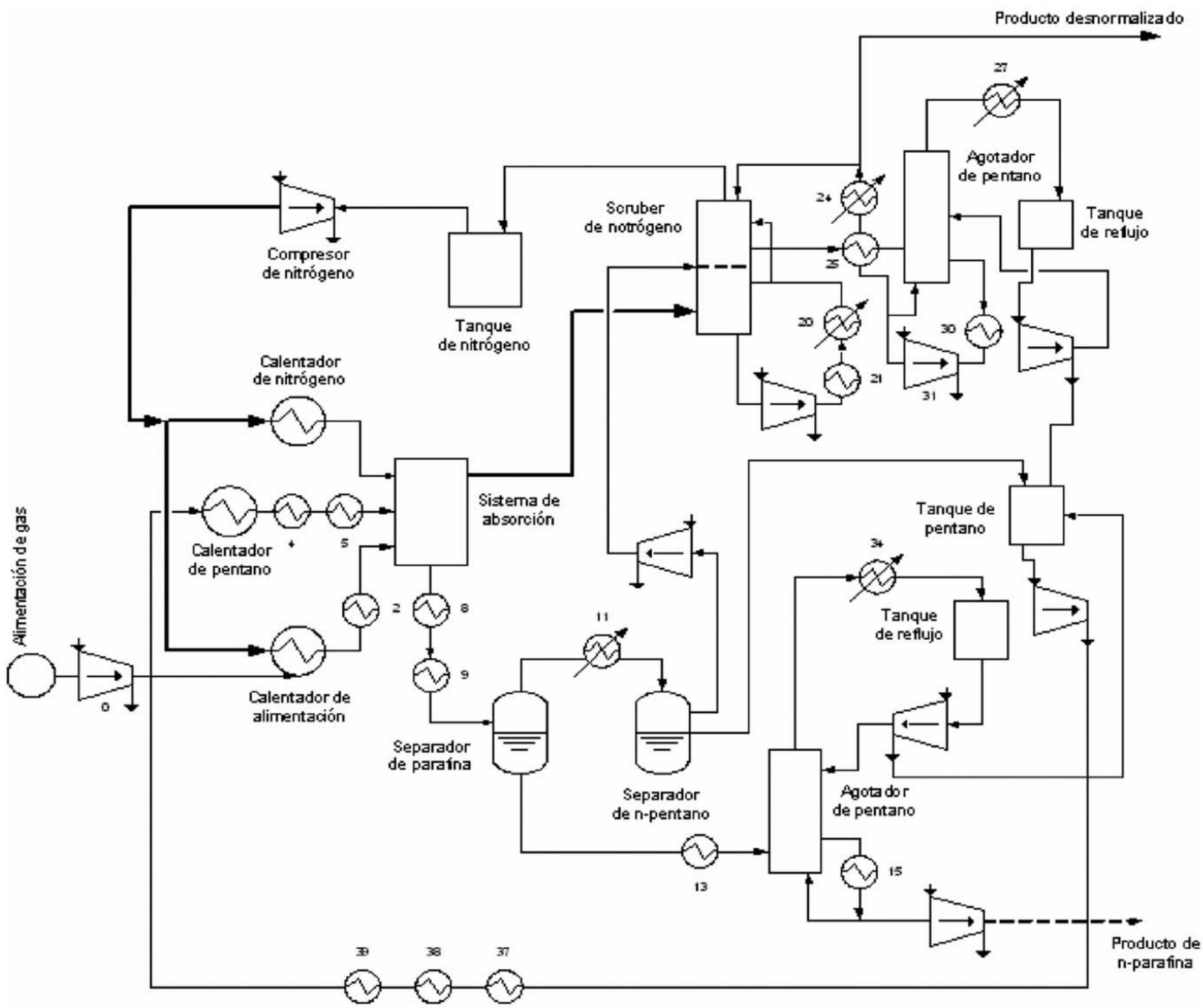


Fig. 19. Normal paraffin extraction.

They were presented with 11 systems of different characteristics and dimensions to evaluate AGE performance (all the systems presented in this paper plus additional ones). There were also given a questionnaire to assess the utility of AGE and understandability of the different produced explanations. Although AGE has been only assessed by a small group, it received very positive and encouraging comments.

Explanations were produced on line, with exception of some behavioral graph that were huge. These graphs were generated before and store in files.

The people who evaluated AGE were first introduced to the project's objectives, how to use AGE and the evaluation objectives. The users were exposed to a tank system and the AGE's help facilities (shown in Fig. 22).

To analyze the questionnaire results we use an interval of confidence of 95% for the proportions of each modality in each question. The formulas utilized was $(1-\alpha)100\%$ for each p_i: $\hat{p}_i \pm z_{\alpha/2}\sqrt{\hat{p}_i(1-\hat{p}_i)/n}$, if $\alpha = 0.05$ then $z_{\alpha/2} = z_{0.025} = 1.96y\,\hat{p}_i = n_i/n$, where n_1, n_2, n_3, n_4 represent counts for category and $n = n_1 + n_2 + n_3 + n_4$ (Mendehnall and Sincich, 1997).

The utility for the students and chemical engineerings are shown in Fig. 23(a) and (b), respectively. Considering 1 for Nothing, 2 for Little, 3 for Regular and 4 for Much.

The statistical data are shown in Fig. 24.

3.2.1. Utility evaluation

In the evaluation, with a confidence interval of 95% between 56 and 87% of the people selected the option "Very", and between 7 and 36% "regular", they selected the option "More of less".

The majority of the people consider that AGE has utility to chemical engineers and all people asked considered it useful to students.

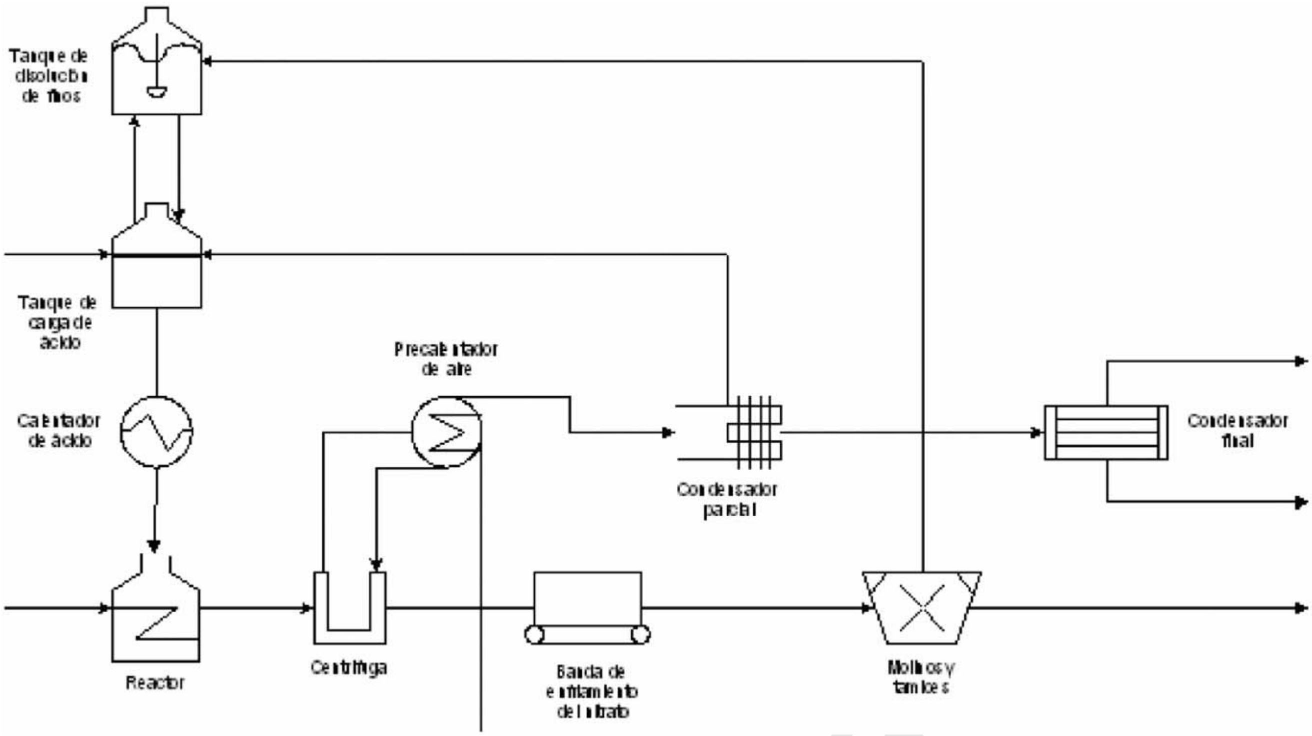


Fig. 20. Production of nitric ammonium.

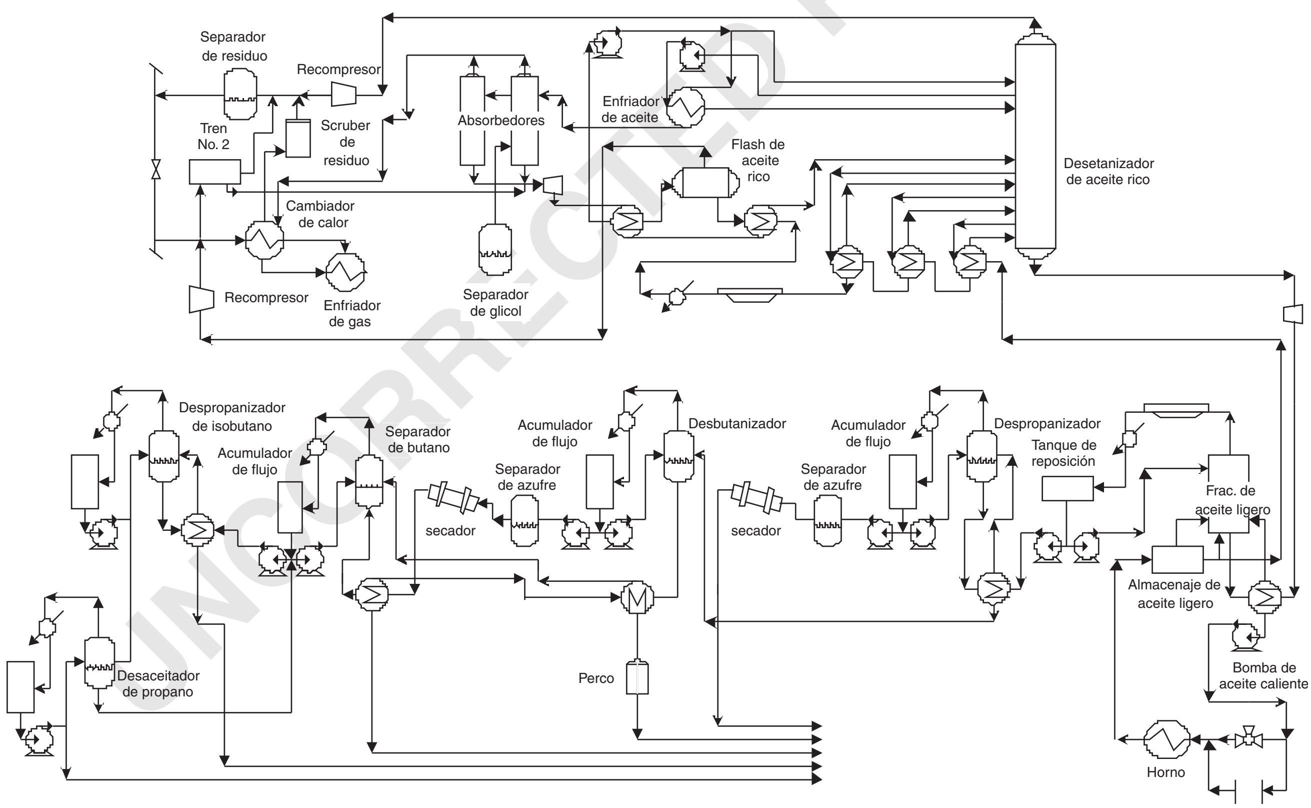


Fig. 21. Empress chemical plant.

3.2.2. Useful evaluation

The explanations considered most useful were the behavioral (between 72% and 97% consider "very useful"), then the functional explanations (between 64% and 92% consider "very useful") and finally the causal

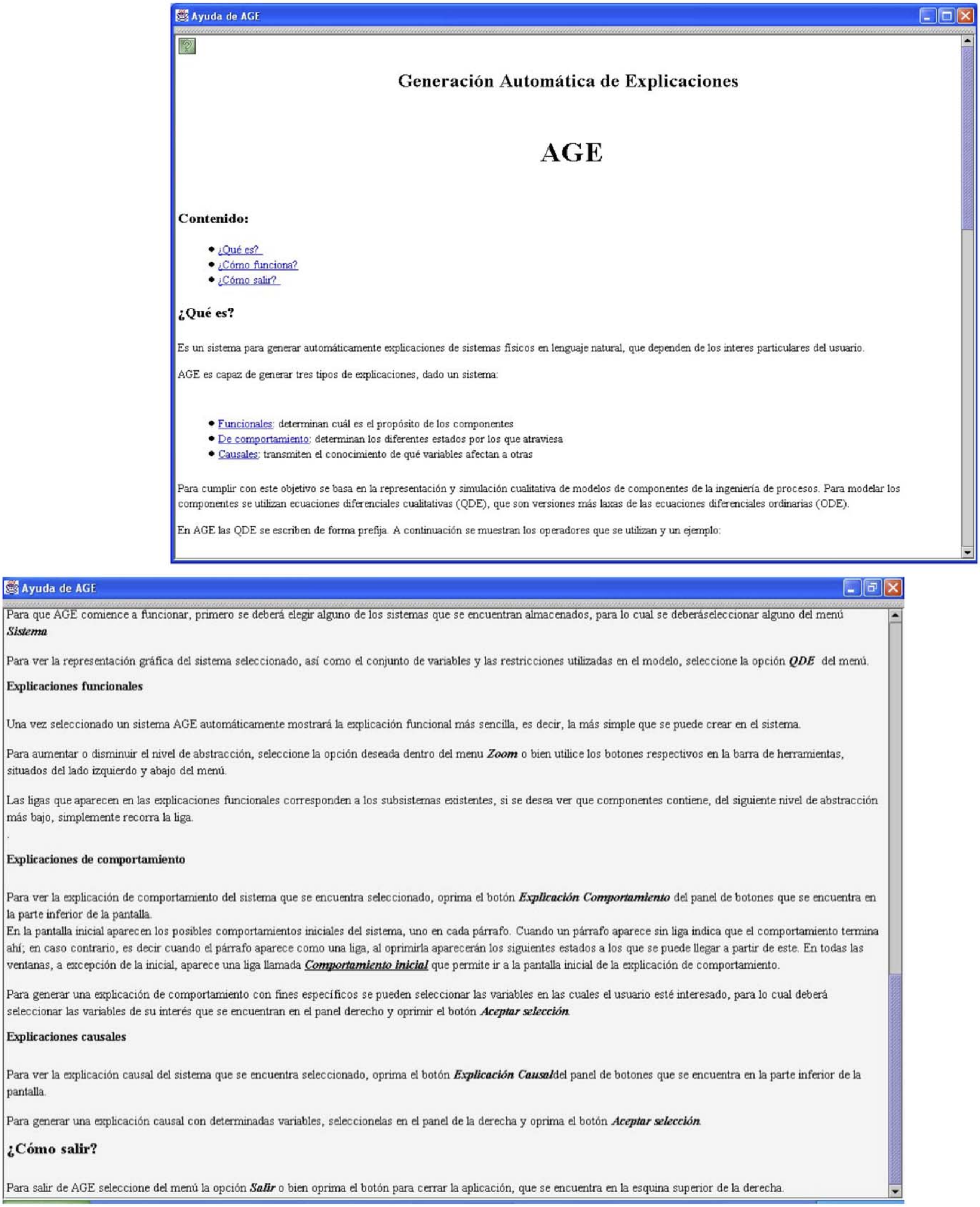


Fig. 22. Help of AGE.

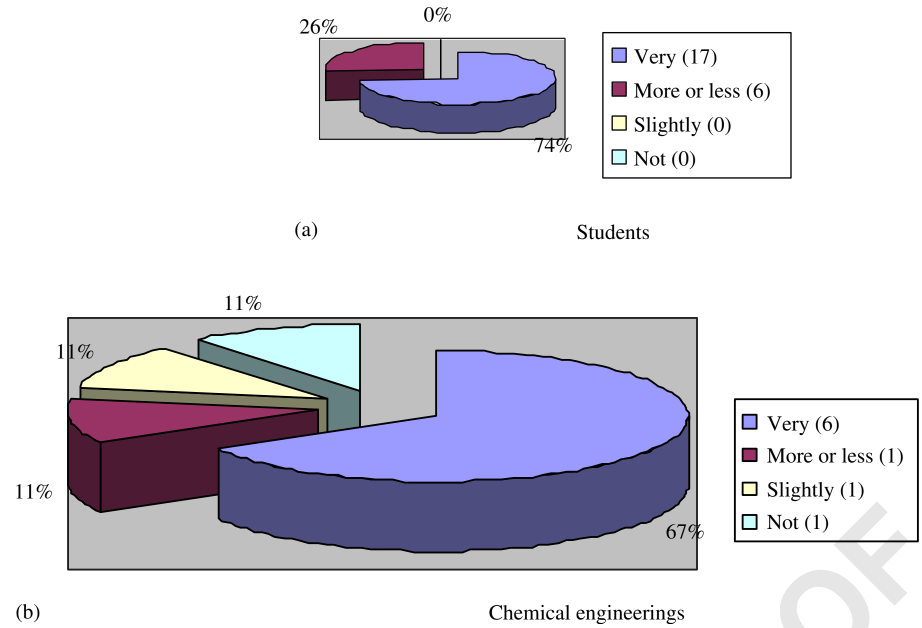


Fig. 23. Utility of the system AGE for students and chemical engineering.

Concept	Value
Median	4
Mode	4
Rank	3
Confidence interval for "Not"	0.03125 +/- 0.06028537
Confidence interval for "Slightly"	0.03125 +/- 0.06028537
Confidence interval for "More or less"	0.21875 +/- 0.14323532
Confidence interval for "Very"	0.71875 +/- 0.15578164

Fig. 24. Statistical data for the utility of AGE.

explanations (60% and 90% consider "very useful") all of them with an interval confidence of 95%.

4. Related work

Although, there have been several related proposals in the literature to produce explanations of physical devices, none produces causal, behavioral and functional explanations in natural language. In a position paper, Bouwer and Bredeweg (1999) argue about the need to take into account techniques from natural language processing and intelligent tutoring systems to improve the production of meaningful explanations from qualitative reasoning. They argue that explanations required knowledge about the tasks and goals of the user and that in general it is not enough to describe a process but also to identify a concept or to contrast it to another. Other authors, like Forbus (1996), have used qualitative reasoning for educational purposes. They use *active illustrations*, that allow a student to modify parameters and relationships of a qualitative model and obtained an explanation summary of the behavior. This approach is focused for middle-school education and cannot produce explanations at different abstraction levels, however, it allows active exploration by the student that can help to improve the understanding of different physical phenomena.

In Chong (1995) a system is described which is used to determine the functionality of a device. It is, however, restricted to function identification, it is unable to handle cycles and does not produce explanations in natural language.

A more recent, although similar system, has been developed by Mizoguchi et al. (Kitamura et al., 2002; Sasajima et al., 1995). They use an ontology and a function and behavior representation language to describe the behavior and functionality of a device using also text templates. Their work, however, does not produce explanations in natural language, it is restricted to function identification, does not consider sub-systems, and is restricted to thermodynamics.

CyclePad (Forbus et al., 1999) was created to analyze and design thermodynamic cycles. It also uses compositional modeling, performs constraint propagation over numerical models, and responds in natural language to questions related with design of thermodynamic systems and values of particular variables. CyclePad was created as an aid for engineering students in task related with design, while AGE was created primarily as an aid for explanation to engineering students.

AGE is not restricted to a particular type of explanations and the user is able to define what variables or subsystems to consider to meet her particular needs.

Several improvements have been suggested on QSIM, however, must of them have been oriented towards more efficient filtering mechanism and extensions to combine it with numerical data (Kuipers, 1994, 2001), and little has been done on component decomposition.

DecSIM (Clancy and Kuipers, 1997) is a model decomposition and simulation algorithm that uses a divide and conquer approach. The variables of the system are partitioned into components so that closely related variables are constrained with the same partition describing the relationships between variables with partition. Each component is viewed as a separate system and is simulated using a state-based representation limited to the variables within the component. Interactions between components are reasoned about as needed to constrain each component. Two types of variables are constrained within each sub-model, within-partition and boundary. DecSIM uses QSIM. The principal differences are that AGE divides automatically a system into subsystems, while DecSIM partitions are manually introduced and only simulates individual components, additionally DecSIM also needs simulate the components that share variables.

In terms of dividing systems into subsystems, Chong (1995) finds the system functionality of a chemical processes. The unit representation in based on Chandrasekaran (1996) and uses a functionality precedence to group immediately neighbors. This works is similar to the subsystem reduction algorithm presented by Chong, however, they are not able to consider cycles, that are very important to chemical engineers.

5. Conclusions and future work

This paper has described a system called AGE capable of generating explanations in natural language from different perspectives and at different abstraction levels. In particular, AGE⋯uses qualitative models and compositional modeling to create a qualitative model of an engineering device. The qualitative model is used to create a causal graph, which is used to produce causal explanations. The simulation of the model, using a process to join individual behavioral graphs, is used to produce behavioral explanations. Behavioral graphs are also used to identify particular functions of devices. AGE is able to automatically divide a complex system into subsystems, and produce explanations in natural language using user-selected variables at different abstraction levels.

AGE has been tested on several engineering systems and with several users with very promising results. As part of our future work, we would like to produce explanations in other languages, the most obvious candidate being English, and have a friendly user interface to specify new components into the library.

Also we plan to try our subsystem reduction approach in other domains such as electrical and mechanical and also with the approach of the Qualitative process theory of Forbus (1984) to show the generality of AGE.

6. Uncited References

González-Brambila, 2003; González-Brambila and Morales, 2003b.

References

Bouwer, A., Bredeweg, B., 1999. Explanation and Qualitative Reasoning. In: Price, C. (Ed.), Proceedings of the International Workshop on Qualitative Reasoning (QR-99), pp. 27–31.

Chong, T.T., 1995. Derivation and use of Function in the Design of Chemical Processes. Technical Report M.Sc. Thesis, University of Edinburgh, Edinburgh, UK.

Falkenhainer, F., Forbus, K., 1991. Compositional modelling: finding the right model for the job. Artificial Intelligence 51, 95–143.

Forbus, K., 1996. Self-explanatory simulators for middle-school science eduaction: a progress report. In: Proceedings of the International Workshop of Qualitative Reasoning (QR96).

Forbus, K.D., Whalley, P.B., Everett, J.O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S., 1999. CyclePad: an articulate virtual laboratory for engineering thermodynamics. Artificial Intelligence 114, 297–347.

González-Brambila, S., 2003. Generación Automática de Explicaciones, Ph.D. thesis, Computer Sciences, Instituto Tecnológico y de Estudios Superiores de Monterrey.

González-Brambila, S., Morales, E., 2003a. Explaining how Engineering Devices Work with AGE. In Proceedings of QR2003, 17th International Workshop on Qualitative Reasoning, Brasilia, Brazil, pp. 103–110.

González-Brambila, S., Morales, E., 2003b. Subsystem reduction for qualitative simulation. In: Proceedings of *QR2003,* 17th International Workshop on Qualitative Reasoning, Brasilia, Brazil, pp. 95–101.

Gruber, T.R., Oisen, G.R., 1994. An ontology for engineering mathematics. In: Proceedings of the Fourth international conference on principles of knowledge representation and reasoning, San Mateo, CA, pp. 258–269.

Iwasaki, Y., Simon, H., 1993. Causality and Model Abstraction., Technical Report KSL-89-80 (Revised Version). Stanford University, Knowledge System Laboratory, Stanford, CA.

Kitamura, Y., Sano, T., Nambar, K., Mizoguchi, R., 2002. A functional concept ontology and its application to automatic identification of functional structures. Advanced Engineering Informatics 16 (2), 145–163.

Kuipers, B., 1994. Qualitative Reasoning: Modelling and Simulation with Incomplete Knowledge. MIT Press, Boston, MA.

Sasajima, M., Kitamura, Y., Ikeda, M. and Mizoguchi, R., 1995. FBRL: a function and behavior representation language. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95), pp. 1830–1836, Kaufmann, Menlo Park, CA.