sensors

MDPI

Article

Smart Sensorization Using Propositional Dynamic Logic

Salvador Merino [1,*,†,‡], Alfredo Burrieza [2], Francisco Guzman [3] and Javier Martinez [1,‡]

[1] Department of Applied Mathematics, University of Malaga, 29071 Malaga, Spain; jmartinezd@uma.es
[2] Department of Philosophy, University of Malaga, 29071 Malaga, Spain; burrieza@uma.es
[3] Department of Electrical Engineering, University of Malaga, 29071 Malaga, Spain; f_guzman@uma.es
* Correspondence: smerino@uma.es; Tel.: +34-655-751-259
† Current Address: Faculty of Industrial Engineering, University of Malaga, 29071 Malaga, Spain.
‡ These authors contributed equally to this work.

Abstract: The current high energy prices pose a serious challenge, especially in the domestic economy. In this respect, one of the main problems is obtaining domestic hot water. For this reason, this article develops a heating system applied to a conventional water tank in such a way as to minimize the necessary energy supply by converting it, under certain circumstances, into atmospheric. For this purpose, the domotic system has been equipped with sensors that automate the pressurization of the compartment and solenoid valves that regulate the external water supply. This design, to which different level sensors are applied, sends the information in real time to an artificial intelligence system, by means of deductive control, which recognizes the states of the system. This work shows the introduction of an extension of propositional dynamic logic in the field of energy efficiency. Thanks to this formalism, a qualitative control of the program variables is achieved by incorporating qualitative reasoning tools. On the other hand, it solves preventive maintenance systems through the early detection of faults in the installation. This research has led to the patenting of an intelligent domestic hot water system that considerably reduces energy consumption by setting disjointed heating intervals that, powered by renewable or non-renewable sources, are controlled by a propositional dynamic logic.

Keywords: domotic; logic; qualitative reasoning; solar water heating; domestic hot water

check for updates

Citation: Merino, S.; Burrieza, A.; Guzman, F.; Martinez, J. Smart Sensorization Using Propositional Dynamic Logic. *Sensors* **2022**, *22*, 3899. https://doi.org/10.3390/s22103899

Academic Editor: Manuel Fuentes Conde

Received: 20 April 2022
Accepted: 17 May 2022
Published: 20 May 2022

1. Introduction

In this paper we present an intelligent control system for the storage and supply of domestic hot water suitable for domotic installations. This system can take advantage of alternative energy sources (such as DHW solar panels and/or photovoltaic panels for heating the electrical resistance) in order to achieve the highest degree of sustainability and ecology. The behavior of this system can be represented by a logical formalism that allows to describe the processes of the system, to detect failures in it, and to facilitate the tasks of maintenance of the installation. The formalism used is very versatile and allows its application in very diverse areas of domotics, which we will demonstrate through some examples.

Current domotic control systems [1] and energy efficiency [2] have evolved from the simple activation and deactivation of loads [3] to having to consider combinations of states in sensors and actuators [4], determining the mode of operation of the same. An example of this can be found in the control of large photovoltaic solar farms [5], where possible reconfiguration [6] and maintenance [7] problems in the panels require precise measurement [8] and control of each unit [9] through separate activations or outages [10]. The profusion of states to be considered in some systems means that a process is needed to ensure that none of the circumstances that may occur in that process are ignored.

This is not something new; in fact, there are already languages that try it, such as GRAFCET, using logic gates [11]. This language is a procedure of synthesis and reduction of both the steps and the memory locations needed to automate an industrial process, using the existing digital simplification methods [12]. Its main problem is that if there is a malfunction of the deduced system, it is very difficult to determine where the error is, since the relationship between the resulting equation and the real logic of the system's operation has been lost [13]. We need a formalism in which that does not happen, so that it can easily be determined where the dysfunction of the resulting equation is located. One of our fundamental aspirations in this paper is to introduce a logic that takes this aspect into account.

On the other hand, in domotic systems, many applications require incremental regulation in what are called "relative values" [14]. An example can serve to clarify these concepts: if we have an intelligent irrigation system, its operation will allow irrigation of it only when the humidity level of the soil requires it, and it can even take measurements in several locations to determine an average value of that humidity [15]. If the value of this humidity advises it, the irrigation system will be activated, but how long will it be irrigated? Normally, for a complete irrigation cycle to be carried out, it suffices that the activation threshold value has been reached, whether or not so much water is required. We need a reasoning system through which we can get it to irrigate, not according to a specific calendar and time, but in the fair value of humidity that our plants require [16]. For this reason, what we will undertake is to increase "a little" the necessary activation time until the desired humidity level is reached.

The examples presented prove the importance of considering qualitative aspects that require to have an intelligent control. The advantage of using qualitative information is that it allows us to handle situations in which we do not need precise numerical information of the system to make decisions [17]. In doing so, we incorporated tools of the qualitative reasoning (QR) to domotic, an approach with many applications in artificial intelligence (AI) (see, for example, [18]).

We propose in this paper a formal representation of domotic systems dealing with all these issues and applied to our invention. Our purpose is to overcome both errors and omissions that may have occurred during its application, in order to be able to contrast what was programmed with the intended rational operation. In addition, the formalism employed allows the incorporation of qualitative information on the status of certain variables and manages it to perform actions [19].

A domotic system can be modeled as a (finite) set of objects belonging to a building (windows, lights, taps, awnings, elevators, appliances, heating systems, etc.) or physical entities with measurable magnitudes (temperature of the water or of a room, the water flow of a tap, the intensity of light, etc.). We will express the properties of states by labels (state labels) that can be numerical (if we want to be precise) or qualitative. Qualitative state labels can be either dichotomous (e.g., open or closed) as well as with gradations (e.g., little, quite, many, . . .) [20].

The representation of the behavior of a domotic system can be performed by what are called *scenes*. A scene is an abstract representation of the behavior of a domotic system at a given time, taking into account the state of the variables that we use to describe it. Since a domotic system is a dynamic system, it undergoes modifications, which gives rise to a transition between scenes. In our approach, this transit can be expressed through operations of composition between qualitative labels as values of our variables. For this purpose we will use a qualitative arithmetic based on absolute orders of magnitude (AOM) [21], in which different qualitative sum operations will be defined. The orders of magnitude arise from making some kind of partition of the real line into equivalence classes, where the numbers are grouped under the same category or qualitative class (large, medium, small, . . .). Arithmetic operations between qualitative classes depend on how the boundaries of these intervals are defined. A study of the properties of different tables for the qualitative sum can be found in [22].

An appropriate formalism to attend transitions between states as well as qualitative arithmetic operations which can generate these states is propositional dynamic logic (PDL) [23]. An advantage of using PDL as a base is that we can make the transition between scenes using a program. PDL also allows us to describe the interaction between actions of computer domain and propositions about situations that we consider relevant for the system that we are dealing with. In addition, it is easy to integrate qualitative arithmetic into PDL through programs. The extended PDL with this feature will be called *Dom*.

Other logics are also designed to deal with state transitions (e.g., transaction logic [24], event calculus [25], situation calculus [26]); however, we do not need represent specifically the time (as event calculus or situation calculus). In addition, many of the aforementioned are predicate languages [27], which leads to problems with decidability. In our case we have a decidable propositional language, PDL [28], for which has been developed *model checking* techniques [29] and *automated theorem proving* techniques [30] through application examples [31].

The article is structured as follows: Section 2 introduces the intelligent multi-behavioral solar water heating system. Section 3 is dedicated to the logic *Dom*, created specifically to solve domotic systems. In Section 4, the formalization of the intelligent solar water heating system is carried out, using propositional dynamic logic. The last section is devoted to conclusions.

2. Design of an Intelligent Solar Water Heating System (ISWHS)

The system presented is a hot water accumulator model adapted for solar collector systems which also incorporates an electrical resistance to heat the water of the accumulator when it is required. In actuality, there are multiple systems of this type and in all cases the consumption of hot water is replaced by cold water entering from the external network causing a decrease in temperature, while the electrical resistance tries to compensate for this loss of heat. The problem is that it is not always possible to maintain a temperature acceptable to the user, and, in addition, there is a significant consumption of electrical energy.

Our system solves these problems, is innovative in its design and operation (there is a patent: [32]), minimizes the consumption of electrical energy, and ensures the maximum efficiency of solar panels. Briefly stated, we start from an initial situation in which the accumulator is completely full, the inlet valve of cold water from the network is open, and the air inlet/outlet valve and hot water outlet valve are closed. When there is no solar supply and there is hot water consumption, the system will allow, by means of a set of solenoid valves (the air inlet/outlet valve and the hot water outlet valve open, meanwhile the cold water closes), that the accumulator is emptied to a certain level, significantly lower than the maximum (*first stage*). At that moment, the *second stage* begins, during which cold water is allowed to enter from the network and an electrical resistance can be placed into operation. When the water reaches a predetermined intermediate level (*third stage*), the cold water valve closes. With the consumption, the water of the accumulator descends again to the low level (of the second stage) and the process is repeated, oscillating the water between the two levels (low and intermediate). This process continues as long as there is no caloric inlet from the solar panels. When the solar collector circuit supplies hot water again, the cold water circuit is opened and the electrical resistance (*fourth stage*) is deactivated. When the water reaches the maximum level (*fifth stage*), the system returns to the initial state (closing the air inlet/outlet valves and the hot water inlet valve).

We note that the fourth stage can begin in any of the first three stages above, depending on how long the solar collectors circuit is inactive. The second hot water outlet (⑦ in

Figure 1) is opened and closed automatically and its behavior has not been described, as it lacks logical control.

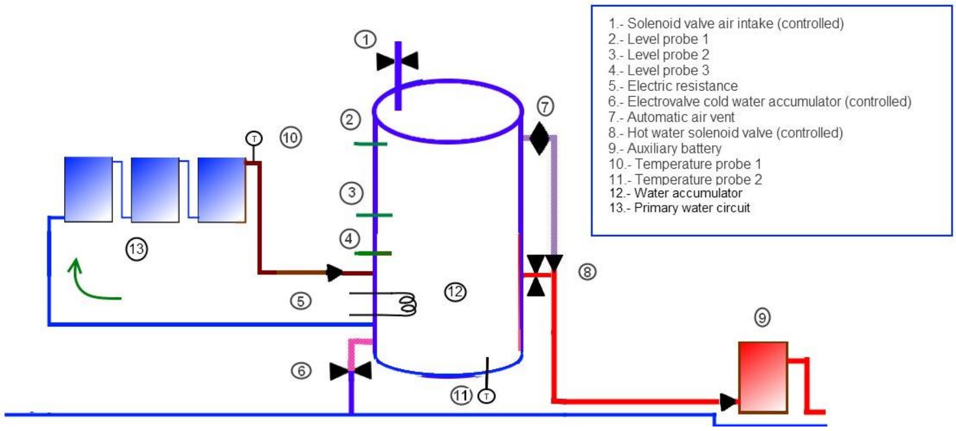

Figure 1. Components description.

This system has several advantages over traditional systems:

- In traditional systems, the cold water that enters from the network is heated to replenish the hot water consumed when there is not enough solar radiation. In our case, the accumulator is left emptying and the only contribution of the electrical resistance is usually used to keep the water in the accumulator hot (but at a low level of the accumulator volume). Once the solar contribution is recovered, cold water is introduced from the network but it is heated by solar energy. This is a significant energy saving.
- As no cold water is introduced during the first stage, in which the water of the accumulator is emptied by the consumption of hot water by the users, there is no decrease of the water temperature in the accumulator by the inlet of cold water, which is normal since the cold water inlet valve is closed. This system keeps the water hot *for longer*.
- During the second stage, the hot water consumed is replaced by cold water from the network heated by the electrical resistance. However, since the water volume of the accumulator is lower (not the entire tank, but a quarter), the heat input of the resistance is also lower than a traditional system. Therefore, the energy saving also occurs at this stage.
- During the third stage, the accumulator is filled with cold water from the network to return to the initial situation and the solar collectors are heating the water. The temperature of the water inside the accumulator decreases and therefore the temperature difference in relation to the water temperature of the solar collectors is maximum. For this reason, this design ensures maximum efficiency of the solar panels, since the differential of temperature of the water passing through the panels with the water of the accumulator will always be maximum. This produces the maximum performance of the solar collectors.

2.1. System Description

The system presented (see Figure 1) aims to achieve maximum energy efficiency. Unlike traditional ones, where the accumulator has four inlets/outlets, the current one has six. The conventional inlets/outlets are those that connect the accumulator ⑫ to the solar collectors or primary circuit ⑬, a cold water inlet of the network controlled by the solenoid valve ⑥ and a hot water outlet controlled by the solenoid valve ⑦. The two additional inlets/outlets are one to allow air to enter the tank ① and another one for the hot water outlet ⑧. The solenoid valves or non-return valves have the purpose of controlling the opening/closing of the inlets/outlets. In total there are three solenoid valves, two of which we have already mentioned, and are as follows:

- The air inlet to the accumulator (A).
- The cold water inlet from the network to the accumulator (CW).
- The hot water outlet from the accumulator (HW)
  (located in the figure in the places ①, ⑥, and ⑧, respectively).

We also have three water level probes inside the accumulator (numbers ②, ③, and ④ and an electrical resistance (ER) ⑤).

In this system we will consider three types of temperature:

- The temperature of the water inside the accumulator (AT), measured by the temperature probe ⑪.
- The temperature of the water passing through the panels (PT) measured by the temperature probe ⑩.
- The temperature requested by the user or *setpoint* temperature (ST).

2.2. System Operation

The general operation is based on the comparison between three different types of temperature and the levels reached by the water inside the accumulator, which will condition the opening and closing of three solenoid valves and the activation or deactivation of an electrical resistance.

We start with a situation where the accumulator is full of water and the CW is open while A and HW are closed.

1. The operation of the primary circuit is automatic: when the temperature of the water leaving the panels (the same water passes through them, one after another, since they are in series) is greater than that of the water inside the accumulator, water circulates due to the thermosyphon effect. When this is not the case, the water will not flow. This does not need control since it is a physical phenomenon.
2. Let us describe the operation of the rest of the circuit:

   - When AT is greater than PT, the three solenoid valves will be activated as follows: CW closes, while A and HW open.
     The three valves remain in their respective states until the water level inside the accumulator reaches the position of the level probe 3 (LP3) (see Figure 2).
     At that time, CW opens and we proceed to activate, or not, the electrical resistance (ER), according to the comparison between AT and ST as follows:

     - If AT is lower than ST, then ER is activated to the *hot* state;
     - If AT = ST, then ER is activated to the *normal* state;
     - if AT is greater than ST, then ER must be deactivated (*cold* state).

     In addition, cold water will continue to enter until the water level inside the accumulator reaches the level probe 2 (LP2). In this case, CW is closed.
     This process will allow the water level to be between probe 2 and 3 while the hot water is being used to consume through the lower outlet of the accumulator (see Figure 3).
   - When PT is greater than or equal to AT, CW will open and ER will be deactivated; when the water reaches level probe 1 (LP1), A and HW will close (see Figure 4).

   It should be noted that this process will lead to AT being greater than or equal to PT (again, previous case). Recall that previously we started from the situation we have now reached with respect to the three solenoid valves: CW is open while A and HW are closed; this describes a circuit.

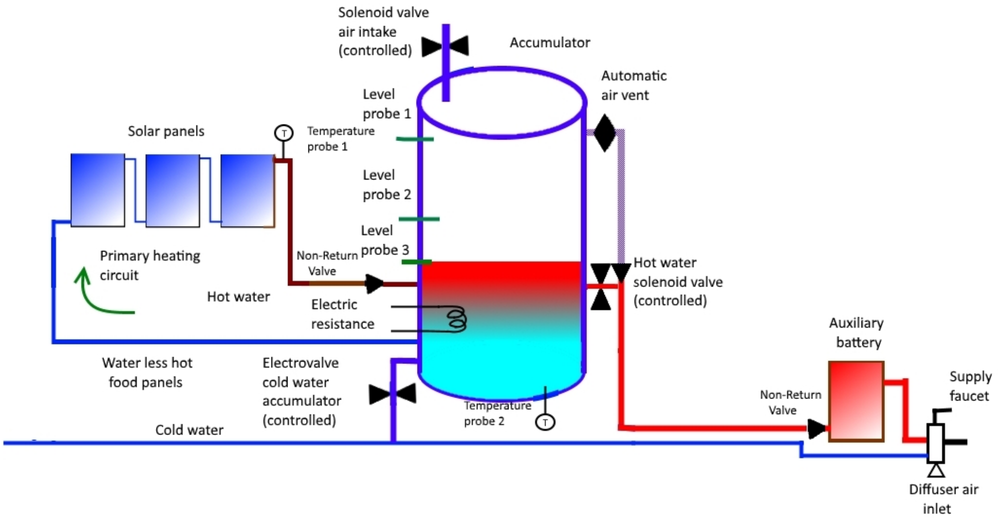

Figure 2. Functioning scheme 1.

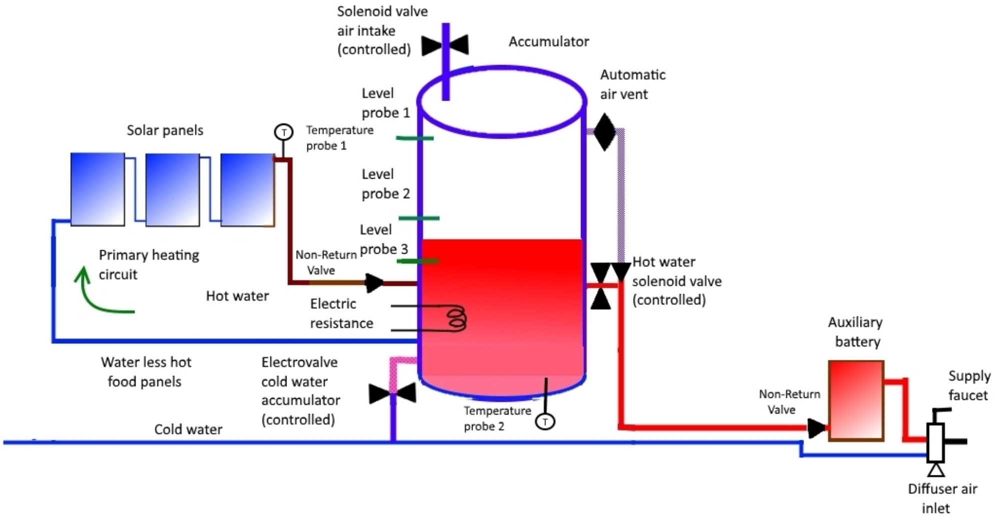

Figure 3. Functioning scheme 2.

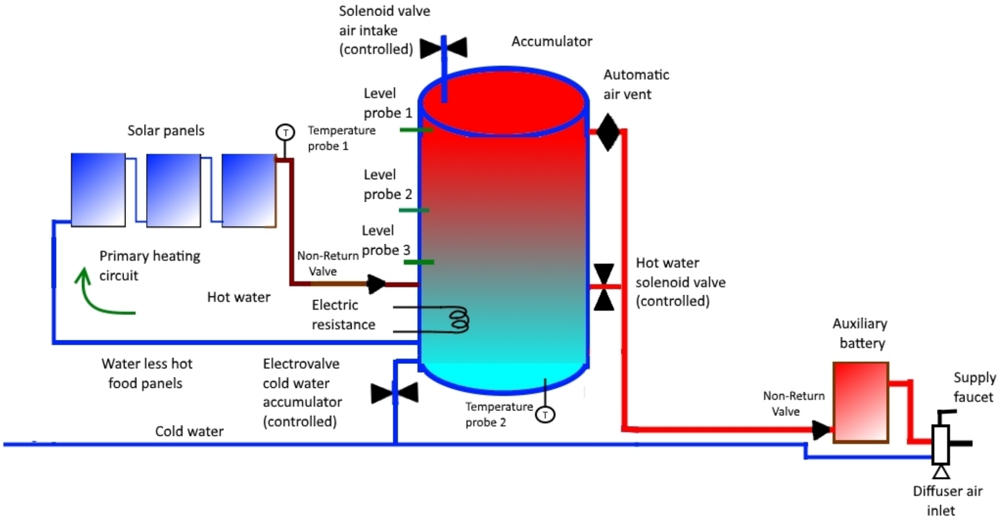

Figure 4. Functioning scheme 3.

With this system we can guarantee the maximum efficiency of the solar panels, since the temperature differential with the water inside the accumulator will always be maximum.

In addition, the cooling of the water inside the accumulator is also prevented by the entry of cold water from the network, a situation that occurs when hot water is consumed in conventional installations.

The operation of the above described system can be expressed by the following Algorithm 1:

Algorithm 1 Procedure of an intelligent water heating system (PIWHS)

Initial situation: the accumulator is full of water, CW is open, and A and HW are closed.

```
begin    // Compare AT and PT //
while true do
  if AT $>$ PT then
      Close CW and Open A and HW;
      if the water level of the accumulator reaches the probe 3 then
        Open CW;
        if
          AT $<$ ST $\Rightarrow$ pass ER intensity to hot;
          AT = ST $\Rightarrow$ pass ER intensity to normal;
          AT $>$ ST $\Rightarrow$ Turn off ER; // ER must be in cold state //
        fi
      end if
      if the water level of the accumulator reaches the probe 2 then
        Close  CW;
      end if
  else       // AT $\leq$ PT //
      Open CW and Turn off ER;
      if the water level of the accumulator reaches the probe 1 then
          Close A and HW;
      end if
  end if
end while
end
```

3. The Logic *Dom*

When using tools for solving domotic problems, such as the one described above, the application of propositional dynamic logic will be used. In this case it will be called *Dom* logic.

3.1. Notation and Basic Concepts

We will use two types of expressions. On the one hand, expressions such as $(x_1, \ldots, x_k)$ are called *scene formulas*, where each x_i is an element of a set of state labels L_i. These labels express gradations of the physical magnitudes (e.g., light intensity, tap water flow, room temperature, etc.) or certain states in which the element of the system are found (e.g., the elevator is on second floor, the tap is closed). For example, consider a domotic system in which we want to refer to the state (off or on) of the light and the degree of temperature of a room. We can consider a scene $x = (x_1, x_2)$, where x_1 is the state of light within the range $L_1 = \{\mathsf{on}, \mathsf{off}\}$ and x_2 is the state of the room temperature within the range $L_2 = \{\mathsf{very\ low}, \mathsf{low}, \mathsf{normal}, \mathsf{high}, \mathsf{very\ high}\}$. Then the scene $(\mathsf{on}, \mathsf{high})$ means that the light is on and the room temperature is high. In actuality, in a scene formula we wish to represent only certain relevant aspects of the states in which a domotic system is found; however, many other things can happen that are beyond the description of the scene formula and which, eventually, we may be interested in taking into account. That expressed by the previous scene $(\mathsf{on}, \mathsf{high})$ could happen, for example, whether the room temperature is pleasant for the homeowner or not, as if one of the lamps does not work, or all are in perfect condition, etc.

In a scene formula we can use a special label, the *empty label*, denoted by "$-$"; its intuitive meaning is to represent that there is no information about the state of a particular component of the scene or that it is not necessary to specify that state (because it has not been modified from a previous specification or for any other reason). The label "$-$" also allows us to create a common scene to specific scenes, which implies a greater degree of abstraction. Thus, $(\mathsf{on}, -)$ refers as concrete instances to $(\mathsf{on}, \mathsf{very\text{-}low})$, $(\mathsf{on}, \mathsf{low})$, etc. Note that "$-$" (which is not actually a state label) in the latter sense comes to play the logical role of a variable.

Given a scene formula $x = (x_1, \ldots, x_k)$, we will use the expression $x^{\{\ell_1/1, \ldots, \ell_k/k\}}$ to indicate that we replace *simultaneously* in x each component x_i for ℓ_i (we admit the possibility $x_i = \ell_i$. This indicates that there is no substitution in the i position. It will be useful to consider this possibility), which we denote abbreviated by ℓ_i/i and where ℓ_i denotes a state label in L_i. The expression $x^{\{\mathfrak{p}_1, \ldots, \mathfrak{p}_r\}}$ means that we highlight in x the positions $\mathfrak{p}_1, \ldots, \mathfrak{p}_r \in \{1, \ldots, k\}$. Similarly, the expression $x^{\{\ell_{\mathfrak{p}_1}/\mathfrak{p}_1, \ldots, \ell_{\mathfrak{p}_r}/\mathfrak{p}_r\}}$, with highlighting positions $\mathfrak{p}_1, \ldots, \mathfrak{p}_r$, is obtained by replacing *simultaneously* the label of each position $\mathfrak{p}_i$ of x by $\ell_{\mathfrak{p}_i}$.

The composite operations to make transitions among scenes are performed between the elements of each set of state labels L_i with a special set of labels called *gradient labels*, denoted by L_G. We can use different sets of gradient labels for different set of state labels depending on the application in mind. As an example of set L_G of qualitative labels, a classical partition of $\mathbb{R}$ in qualitative classes appears in Figure 5 (see [20,21]):

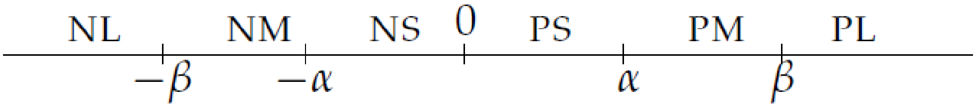

Figure 5. Partition of real line in qualitative classes.

$$\begin{array}{lll}
\textsc{nl} = (-\infty, -\beta) & & \textsc{ps} = (0, \alpha] \\
\textsc{nm} = [-\beta, -\alpha) & 0 = [0] & \textsc{pm} = (\alpha, \beta] \\
\textsc{ns} = [-\alpha, 0) & & \textsc{pl} = (\beta, +\infty)
\end{array}$$

The labels means "negative large" (NL), "negative medium" (NM), "negative small" (NS), "zero" (0), "positive small" (PS), "positive medium" (PM), and "positive large" (PL). The constants α, β are real numbers that are used to delimit the equivalence classes (the particular criteria to choose these numbers would depend on the application in mind).

Now, we introduce some notation. If ℓ_1 and ℓ_2 are two labels (of state or gradient), then $\ell_1 \ll \ell_2$ means that ℓ_1 is previous to ℓ_2, that is, for any $x \in \ell_1$, and $y \in \ell_2$, we have that $x < y$. In the previous case, we have the order $\textsc{nl} \ll \textsc{nm} \ll \textsc{ns} \ll 0 \ll \textsc{ps} \ll \textsc{pm} \ll \textsc{pl}$. When we use state labels that express dichotomy such as on, off, we will consider $\mathsf{off} \ll \mathsf{on}$. The notation $\gg$ indicates the inverse relationship of $\ll$; also, $\ell_1 \underline{\ll} \ell_2$ means that $\ell_1 \ll \ell_2$ or $\ell_1 = \ell_2$.

A set of gradient labels L_G is intended to increase or decrease the magnitudes of the variables of a scene operating on the status labels using composition. Thus, using the previous partition as set of gradient labels, if the light has a *low* intensity (state label), we can increase it a *little*, that is, with a small number or belonging to PS (gradient label), which may result that the light remains within the range of *low* (even if it has increased in intensity) or pass to the *normal* state (new label status). In order to perform the compositing operations in this case, we will translate the state labels into gradient labels and use a previously defined qualitative sum operation according to our interests.

There are several types of expressions that we can form when performing operations of composition in the positions of a scene formula. Therefore, the notation $\mathfrak{p}_i + \ell_i^+ : \mathfrak{p}_i$ indicates that in the position $\mathfrak{p}_i$ we must perform a composition operation (sum) between the state label $\ell_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ and the (positive) label ℓ_i^+. The nature of + (numerical or qualitative) obviously depends on the labels involved. The sum is qualitative if at least one of the factors is qualitative. If both labels are qualitative, then ℓ_i^+ is always a (positive) *gradient* label and the operation must always generate a qualitative state label $\ell'_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ such that

$\ell_{\mathfrak{p}_i} \lll \ell'_{\mathfrak{p}_i}$. If $\ell'_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ is qualitative and ℓ_i^+ is numerical, then the result of the composition is also a qualitative state label $\ell'_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ such that $\ell_{\mathfrak{p}_i} \lll \ell'_{\mathfrak{p}_i}$. In the same sense, it occurs if $\ell'_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ is numerical and ℓ_i^+ is qualitative.

Similarly, $\mathfrak{p}_i + \ell_i^- : \mathfrak{p}_i$ indicates that in the position $\mathfrak{p}_i$, we compose its corresponding state label $\ell_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ with the (negative) gradient label ℓ_i^-, which generates a state label $\ell'_{\mathfrak{p}_i} \in L_{\mathfrak{p}_i}$ such that $\ell'_{\mathfrak{p}_i} \lll \ell_{\mathfrak{p}_i}$.

3.2. Syntax

The language of *Dom* is an extension of the language of PDL with a new set of atomic formulas and specific atomic programs. Concretely, *Dom* contains the propositional operators $\rightarrow$ (material implication) and $\perp$ (falsity), the program operators ; (composition), $\cup$ (choice), and * (iteration) and the mixed operators [ ] (necessity) and ? (test). As defined symbols we have $\top$ (truth), $\wedge$ (and), $\vee$ (or), $\leftrightarrow$ (if and only if), and $\langle\ \rangle$ (possibility) as usual. Formulas and programs are built inductively, using the previous operators, from $\mathbb{P}$ (a set of alphanumeric chains), $L = L_1 \times \cdots \times L_k$ (a set of scene formulas) being each $L_i (1 \leq i \leq k)$ a finite set of state labels and a specific parametrized program (called *regulator*) consisting of several subroutines that analyze below (although *regulator* can be declared in subprograms we consider it an atomic program, since it requires the simultaneity of all the subprograms that form it, so that it can not be obtained from these with the classic PDL operations as ; or $\cup$).

If $x = (x_1, \ldots, x_k)$ is a scene formula, then we have as programs the following ones:

- $\mathtt{Inc}_x^{\{\mathfrak{p}_1+\ell_1^+ : \mathfrak{p}_1, \ldots, \mathfrak{p}_n+\ell_n^+ : \mathfrak{p}_n\}}$ (*increase*): This program means that we must increase the level of the values of the positions $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ of x, respectively, by the positive labels $\ell_1^+, \ldots, \ell_n^+$, using a sum operation + (qualitative or numerical). A simultaneous replacement of $\mathfrak{p}_i$ by $\mathfrak{p}_i + \ell_i^+$ is carried out; e.g., increase a *little* the temperature of the living room.
- $\mathtt{Dec}_x^{\{\mathfrak{p}_1+\ell_1^- : \mathfrak{p}_1, \ldots, \mathfrak{p}_n+\ell_n^- : \mathfrak{p}_n\}}$ (*decrease*): This program means that we must decrease the level of the values of the positions $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ of x, respectively, by the negative labels $\ell_1^-, \ldots, \ell_n^-$, using a sum operation + (qualitative or numerical); e.g., decrease *much* the intensity of the kitchen light.
- $\mathtt{Change}_x^{\{\ell_{\mathfrak{p}'_1}/\mathfrak{p}_1, \ldots, \ell_{\mathfrak{p}'_n}/\mathfrak{p}_n\}}$ (*change*): This program means that we must replace *simultaneously* the state labels of the positions $\mathfrak{p}_1 \ldots \mathfrak{p}_n$ of x (i.e., $\ell_{\mathfrak{p}_1}, \ldots, \ell_{\mathfrak{p}_n}$), respectively, by the state labels $\ell_{\mathfrak{p}'_1}, \ldots, \ell_{\mathfrak{p}'_n}$ (e.g., set the room temperature to 20 °C, ...).
- $\mathtt{Watch}_x^{\{\mathfrak{p}_1, \ldots, \mathfrak{p}_n\}}$ (*watch*): This program means that we simply *watch* the state labels of the positions $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ of x; it is assumed that these would be maintained if there are no actions that change them. The purpose of this program is to explicitly check that these labels do not change even if those in other positions do.

The following program collects all of the above in a single order and is defined:

- $\mathtt{Reg}_x^{\Psi_1, \Psi_2, \Psi_3, \Psi_4}$ (*"regulator"*), where

  $$\begin{aligned}
  \Psi_1 &= \{\mathfrak{p}_1 + \ell_1^+ : \mathfrak{p}_1, \ldots, \mathfrak{p}_l + \ell_l^+ : \mathfrak{p}_l\} \\
  \Psi_2 &= \{\mathfrak{q}_1 + \ell_1^- : \mathfrak{q}_1, \ldots, \mathfrak{q}_m + \ell_m^- : \mathfrak{q}_m\} \\
  \Psi_3 &= \{\ell^\bullet_{\mathfrak{p}'_1}/\mathfrak{p}'_1, \ldots, \ell^\bullet_{\mathfrak{p}'_n}/\mathfrak{p}'_n\} \\
  \Psi_4 &= \{\mathfrak{q}'_1, \ldots, \mathfrak{q}'_o\}
  \end{aligned}$$

  and $\mathfrak{p}_1, \ldots, \mathfrak{p}_l, \mathfrak{q}_1, \ldots, \mathfrak{q}_m, \mathfrak{p}'_1, \ldots, \mathfrak{p}'_n, \mathfrak{q}'_1, \ldots, \mathfrak{q}'_o$ are positions of x, all different, and at least one Ψ_i $(1 \leq i \leq 4)$ is not empty.

That is, *regulator* executes the actions of $\mathtt{Inc}_x^{\Psi_1}, \mathtt{Dec}_x^{\Psi_2}, \mathtt{Change}_x^{\Psi_3}$, and $\mathtt{Watch}_x^{\Psi_4}$ simultaneously, something very important in domotic, and without problems, because the positions of the different subprograms that include it are all different. This allows to create a final scene with all the actions executed of these programs without going through intermediate scenes, which would result in case of proceed sequentially.

Note that we can define the previous programs in terms of *regulator* by a very simple procedure: place the empty set of operations with labels in the corresponding part. Thus, we define

$$\mathtt{Inc}_x^{\Psi_1} =_{def} \mathtt{Reg}_x^{\Psi_1,\varnothing,\varnothing,\varnothing} \qquad \mathtt{Dec}_x^{\Psi_2} =_{def} \mathtt{Reg}_x^{\varnothing,\Psi_2,\varnothing,\varnothing}$$

$$\mathtt{Change}_x^{\Psi_3} =_{def} \mathtt{Reg}_x^{\varnothing,\varnothing,\Psi_3,\varnothing} \qquad \mathtt{Watch}_x^{\Psi_4} =_{def} \mathtt{Reg}_x^{\varnothing,\varnothing,\varnothing,\Psi_4}$$

In general, we will use an expression such as $\mathtt{Inc}_x^{\Psi}$, $\mathtt{Dec}_x^{\Psi}$, etc., when we want to refer to one of these programs without specifying the set of substitutions that the program performs in the x scene. With this in mind, we can introduce new specific programs from $\mathtt{Change}_x^{\Psi}$ to treat dichotomous states, so we have

- $\mathtt{On}_x^{\{\mathfrak{p}_1,\ldots,\mathfrak{p}_n\}}$ ("turn on all the positions $\mathfrak{p}_1,\ldots,\mathfrak{p}_n$ of x").
- $\mathtt{Off}_x^{\{\mathfrak{p}_1,\ldots,\mathfrak{p}_n\}}$ ("turn off all the positions $\mathfrak{p}_1,\ldots,\mathfrak{p}_n$ of x").

  That is, we have
  $\mathtt{On}_x^{\{\mathfrak{p}_1,\ldots,\mathfrak{p}_n\}} =_{def} \mathtt{Change}_x^{\{\mathsf{on}/\mathfrak{p}_1,\ldots,\mathsf{on}/\mathfrak{p}_n\}}$
  and
  $\mathtt{Off}_x^{\{\mathfrak{p}_1,\ldots,\mathfrak{p}_n\}} =_{def} \mathtt{Change}_x^{\{\mathsf{off}/\mathfrak{p}_1,\ldots,\mathsf{off}/\mathfrak{p}_n\}}$.

We can also define *mixed dichotomous programs* that combine the actions of the previous ones. For example,

- $\mathtt{Onoff}_x^{\Psi_1,\Psi_2}$. This program combines $\mathtt{On}_x^{\Psi_1}$ and $\mathtt{Off}_x^{\Psi_2}$, where Ψ_1 and Ψ_2 are sets of distinguished positions of different x.
- $\mathtt{Onoff}_x$ ("exchange on and off") is a generic modification of the previous program $\mathtt{Onoff}_x^{\Psi_1,\Psi_2}$, that simply does not require previous identification of the positions to be activated/deactivated (the system reads the status of all positions and performs this order automatically, going through all of them). In actuality, Ψ_1 are all positions in the scene with off and Ψ_2 are all positions in the scene with on. This program *activates* in x all positions that are deactivated and at the same time *deactivates* all positions that are enabled; e.g., one result would be to turn all lights on (that are off) and turn off all lights (that are on) simultaneously.

Now we give some definitions about scene formulas.

Definition 1. *Let $x = (x_1,\ldots,x_k)$ and $y = (y_1,\ldots,y_k)$ be scene formulas. We say that x is different from y if $x_i \neq y_i$ for some $i \in \{1,\ldots,k\}$.*

Definition 2. *Let $x = (x_1,\ldots,x_k)$ and $y = (y_1,\ldots,y_k)$ be scene formulas and let us consider the program $\mathtt{Reg}_x^{\Psi_1,\Psi_2,\Psi_3,\Psi_4}$. We say that y is x-regulated with respect to (abbreviated wrt) Ψ_1–Ψ_4 if for every i with $(1 \leq i \leq k)$ we have one of the following cases:*

- *$y_i = x_i + \ell_i^+$, where $x_i + \ell_i^+ \in \Psi_1$, $+$ is an operation defined for L_i and ℓ_i^+ is a positive label of the set L_G (chosen for L_i).*
- *$y_i = x_i + \ell_i^-$, where $x_i + \ell_i^- \in \Psi_2$, $+$ is an operation defined for L_i and ℓ_i^- is a negative label of the set L_G (chosen for L_i).*
- *$y_i = x_i^{[l_i/x_i]}$, where $l_i/x_i \in \Psi_3$ and $l_i \in L_i$.*
- *$y_i = x_i$ and $x_i \in \Psi_4$.*

Example 1. *Suppose a domotic system is composed of the open/closed properties of a window and its degree of openness. For this, we use two sets of labels: $L_1 = \{\mathit{on}, \mathit{off}\}$ (open/close states) and $L_2 = \{\textit{very-little}, \textit{little}, \textit{normal}, \textit{much}, \textit{very-much}\}$. As a set of gradient labels L_G for L_2 we have*

$$\{\mathrm{NL}, \mathrm{NM}, \mathrm{NS}, 0, \mathrm{PS}, \mathrm{PM}, \mathrm{PL}\}$$

The normal label denotes a previously established standard minimum aperture. The language contains state formulas such as (x_1, x_2), where x_1 refers to the open/closed state of the window and x_2 to the degree of opening of the window. Then the formula

$$[(\mathit{off}, -)?; \mathtt{Change}^{\{on/1, normal/2\}}_{(off,-)}](\mathit{on}, \mathit{normal})$$

indicates that any action consisting of opening the window (initially closed) results in a state in which the window is open with a normal (stipulated or predetermined) degree of aperture. Then, the formula

$$[\mathtt{Inc}^{\{normal + \mathrm{PS}:2\}}_{(on,normal)}](\mathit{on}, \mathit{normal}) \vee (\mathit{on}, \mathit{much})$$

indicates that any action that opens the window a little more will lead to it either remaining within the normal parameter of opening or change to a new state, i.e., to become quite open (this requires us to predefine a matrix where a + operation is handled between the state labels and gradient labels so that normal $+$ PS leads to normal or much). The notation normal+PS:2 indicates that in the second position of the scene (on, normal) its label is replaced by some label resulting from the composition normal $+$ PS.

Now, we will express a more complex situation. Let us consider the atoms p, q with the respective meanings "the heating is on" and "it is economically beneficial". The following program (denoted by a) expresses the action of closing the window, and also, while the heating is on, makes sure the window is closed:

$$(\mathit{on}, -)?; \mathtt{Off}^{\{1\}}_{(on,-)}; (p?; \mathtt{Watch}^{\{1\}}_{(off,-)})^*; \neg p?$$

Note that $(\mathit{on}, -)?; \mathtt{Off}^{\{1\}}_{(on,-)}$ indicates that if the window is open then it closes, and

$$(p?; \mathtt{Watch}^{\{1\}}_{(off,-)})^*; \neg p?$$

is an instruction of type «while». Then the formula $[a]\, q$ means that the result of the previous action is always something economically beneficial.

3.3. Semantics

A *model* $\mathcal{M}$ is a tuple (W, m) where W is a nonempty set of states (called, more specifically, *domotic states*). We will use letters as u, v, w (with or without indexes) as representing states. By convenience, each element $u \in W$ is to be understood as a domotic state labeled by elements of L.

1. The meaning function m is required to fulfill the following:
   - (a) $m(p) \subseteq W$, for every atomic formula $p \in \mathbb{P}$.
   - (b) $m(x) \subseteq W$, for every scene formula $x \in L$ and $m(x) \neq \varnothing$.
   - (c) For all $w \in W$ there exists a scene formula $x \in L$ such that $w \in m(x)$ (all the states of a model are described by a scene formula).
   - (d) For all $w \in W$ and $x \in L$, if $w \in m(x)$, then x is unique, that is, for all $y \in L$ different from x (in the sense of Definition 1) we have $w \notin m(y)$.
2. We define now the semantics of the specific program *regulator*. if $x = (x_1, \ldots, x_k)$ is a state formula, then

- $m(\texttt{Reg}_x^{\Psi_1,\Psi_2,\Psi_3,\Psi_4})(m(x)) \subseteq$
  $\{v \mid v \in m(y)$ and y is x-regulated wrt Ψ_1-$\Psi_4\}$

3. Finally, if φ and ψ are formulas and a, b are programs, then we have the following:
   (a) $m(\varphi \to \psi) = (W \backslash m(\varphi)) \cup m(\psi)$;
   (b) $m(\bot) = \varnothing$;
   (c) $m([a]\varphi) = \{w \in W \mid$ for all $v \in W,$ if $(w, v) \in m(a)$ then $v \in m(\varphi)\}$;
   (d) $m(a \cup b) = m(a) \cup m(b)$;
   (e) $m(a;b) = m(a) \circ m(b) = \{(w, v) \mid$ there exists
   $u \in W$ such that $(w, u) \in m(a)$ and $(u, v) \in m(b)\}$;
   (f) $m(a^*) = m(a)^*$ (reflexive and transitive closure of relation $m(a)$);
   (g) $m(\varphi?) = \{(w, w) \mid w \in m(\varphi)\}$.

   The semantic concepts of *satisfiability*, *valid in a model*, and *valid* are the usual ones.

4. Formalizing ISWHS

In this section we will express the operation of the system using the programming language of the logic.

4.1. System States

We will use scene formulas with seven state variables $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, where

- x_1 denotes the temperature of the water accumulated in the accumulator (AT).
- x_2 denotes the temperature of the water passing through the solar panels (PT).
- x_3 denotes the status of the cold water solenoid valve of the network to the accumulator (CW).
- x_4 denotes the status of the solenoid valve in the upper part of the accumulator (A).
- x_5 denotes the status of the solenoid valve of the lower hot water outlet of the accumulator (HW).
- x_6 denotes the status of the electrical resistance (ER).
- x_7 denotes the set point temperature defined by the user (ST).

We will also use "propositional constants" to denote a certain behavior of the system:

- *si* denotes that the water level inside the accumulator reaches the probe of the upper level i $(i = 1, 2, 3)$.

The sets of state labels $L_1, \ldots, L_7$ are as follows:

- $L_1 = L_2 = \{\mathsf{vc}, \mathsf{c}, \mathsf{n}, \mathsf{h}, \mathsf{vh}\}$.
- $L_3 = L_4 = L_5 = \{\mathsf{on}, \mathsf{off}\}$.
- $L_6 = \{\mathsf{c}, \mathsf{n}, \mathsf{h}\}$.
- L_7 = a finite subset of $\mathbb{N}$ (to choose).

As gradient labels we have the ones established in the partition of the real line made in Section 3.1, namely, $\{\text{NL}, \text{NM}, \text{NS}, 0, \text{PS}, \text{PM}, \text{PL}\}$. The intuitive meanings of the state labels are $\mathsf{vc} = $ *very-cold*; $\mathsf{c} = $ *cold*; $\mathsf{n} = $ *normal*; $\mathsf{h} = $ *hot*; $\mathsf{vh} = $ *very-hot*. We have the following label orders. For $L_i (i = 1, 2)$: $\mathsf{vc} \ll \mathsf{c} \ll \mathsf{n} \ll \mathsf{h} \ll \mathsf{vh}$; for $L_j (j = 3, 4, 5)$: $\mathsf{off} \ll \mathsf{on}$; and for L_6: $\mathsf{c} \ll \mathsf{n} \ll \mathsf{h}$.

4.2. Composition Operations

In this section we introduce the composition operations. We start with the table for the gradient labels (L_G). Recall the partition made in Figure 5, where we consider $\text{NL} = (-\infty, -\beta)$, $\text{PS} = (0, \alpha]$, $\text{NM} = [-\beta, -\alpha)$, $0 = [0]$ and $\text{PM} = (\alpha, \beta]$. The L_G partition can contain a different numerical set appropriate for each application (a subset of $\mathbb{R}$). On the other hand, from now on, we will consider that $\beta = 2\alpha$.

In Table 1, the operation $+_G$ between qualitative classes yields either a qualitative class or a range of qualitative classes $[X, Y]$ (where $X \ll Y$), which we express, indicating all the qualitative classes that compose this interval separated by commas for clarity. For example, NS$+_G$PS provide us, as a result, the interval [NS, PS], comprising NS, 0, PS. On the other hand, we used 1 to denote the interval [NL, PL].

Table 1. Composition of components of L_G.

$+_G$	NL	NM	NS	0	PS	PM	PL
NL	NL	NL	NL	NL	NL, NM	NL, NM, NS	1
NM		NL	NL, NM	NM	NM, NS	NS, 0, PS	PS, PM, PL
NS			NM, NS	NS	NS, 0, PS	PS, PM	PM, PL
0				0	PS	PM	PL
PS					PS, PM	PM, PL	PL
PM						PL	PL
PL							PL

The operation $+_G$ verifies the following: $+_G$ is *consistent* with the usual real sum $+$, that is, for all qualitative classes, X, Y, and all $x \in X, y \in Y$, $X +_G Y$ corresponds to the smallest interval (in the sense of inclusion) formed by qualitative classes containing $x + y$ (see [19,22]). For example, NM$+_G$PM = [NS, PS].

In Tables 1–3, the state labels are translated to gradient labels to make the composition that we make in Table 1. In Table 2 we take vc = NL, c = NM, n = [NS, PS], h = PM, and vh = PL. In Table 3 we take c $=$ [NL, NM], n $=$ [NS, PS], and h $=$ [PM, PL]. Once the composition is completed, we translate the result to state labels by choosing the smallest possible interval of state labels. For example, the composition n $+_2$ PS in Table 3 is, in fact, [NS, PS] $+_G$ PS $=$ [NS, PM]. This result comes from composing, using $+_G$, in each of the classes of the interval [NS, PS] with PS. Now, since the minimum interval in which [NS, PM] is included in terms of the classes in Table 3 is [n, h] = [NS, PL], we finally have this output.

Table 2 describes the behavior of both the water temperature of the solar panels and the water inside the accumulator. The composition Table 3 with the labels of L_6 (Table 2) allows us to calculate both the increase and the gradual decrease of the ER intensity. For instance, we took a scale for L_G where $\alpha = 5$ and $\beta = 10$. Now we consider the following projection of L_G over $L_1(L_2)$ in Table 2: $-\beta$ corresponds to 20 °C, $-\alpha$ with 25 °C, 0 with 30 °C, α with 35 °C and β with 35 °C. Therefore, we have for L_G the following partition NL = $[\ldots, -10)$, NM = $[-10, -5)$, NS = $[-5, 0)$, [0] = 0, PS = $(0, 5]$, PM = $(5, 10]$, PL= $(10, \ldots]$ and for $L_1(L_2)$ the corresponding partition very cold = $[\ldots, 20)$, cold = [20, 25), normal = [25, 35], hot = (35, 40], very hot = $(40, \ldots]$. Now, adding, for example, a small number (say 3) to a normal temperature (n) can lead us to stay within that temperature or go to hot (h), depending on how close the system is to the limit 35.

Table 2. Composition of components of L_1 (L_2).

$+_1$	NL	NM	NS	0	PS	PM	PL
vc	vc	vc	vc	vc	vc, c	vc, c, n	1
c	vc	vc	vc, c	c	c, n	n	n, h, vh
n	vc, c	vc, c , n	c, n	n	n, h	n, h, vh	h, vh
h	vc, c, n	n	n, h	h	h, vh	vh	vh
vh	1	n, h, vh	h, vh	vh	vh	vh	vh

Table 3. Composition of components of L_6.

$+_2$	**NL**	**NM**	**NS**	**0**	**PS**	**PM**	**PL**
c	c	c	c	c	c, n	c, n	1
n	c	c, n	c, n	n	n, h	n, h	h
h	1	n, h	n, h	h	h	h	h

4.3. *Logical Description of System Behaviour*

The *Dom* logic can describe the behavior of the system. For instance, let us assume a state where AT $>$ PT, AT $<$ ST, the valves CW, A, and HW are open, and ER has a normal intensity and water level reaches probe 3 (a state perfectly coherent with a correct run of the algorithm for the system). Thus, $s3$ and the scene $x = (-,-,\mathsf{on},\mathsf{on},\mathsf{on},\mathsf{n},-)$ hold at that state. The procedure establishes then that a new state where the scene $y = (-,-,\mathsf{on},\mathsf{on},\mathsf{on},\mathsf{h},-)$ holds must be reached. If we operate gradually, increasing the intensity a *little* (a small number) until reaching the hot level, the formula

$$(x \wedge s3) \to [(\neg y?; \texttt{Inc}_x^{\{6+\text{PS}:6\}})^*; y?]\, y$$

describes this process.

4.4. *Detecting Failures in System Operation*

In this section we show how our formalism captures what *should not happen* if we want the system described to work correctly.

We will give some examples of significant failures of the system. We will consider transitions from a state w to a state v, placing below each state some formulas true at it. The failure detected is a trouble with a valve or the resistance. In the following we add new atoms of the type sij, with the intuitive meaning "the water inside the accumulator is between the level probes i and j". Now, we list some troubles emphasizing in the scenes only the information more relevant.

The first case in the previous Table 4 states that in the scene $x = (-,-,-,\mathsf{off},-,-,-)$, the component x_2 belongs to a qualitative class lower than the component x_1. At the state w, the formulas x and $s1$ are true. At the state v, the same scenes x and $s12$ are true. The program establishes that A should be opened if AT $>$ PT (i.e., $x_2 \ll x_1$). This means that the scene x must vary the component x_4 to on. However, at state v, valve A stays off. After a sufficient lapse of time the water will be between probes 1 and 2, so the situation at v should be a different scene of x, namely, $(-,-,-,\mathsf{on},-,-,-)$. This means that, according to the behavior of the program at this stage, the formula $\langle \texttt{Onoff}_x^{\{4,5\},\{3\}} \rangle(-,-,-,\mathsf{off},-,-,-)$ must be false at w. Therefore, if the procedure for the system is correct, we can suspect a failure in valve A. The rest of the cases are interpreted in a similar way. In general, for these troubleshooting tasks in a graph, model-checking techniques can be applied.

Table 4. Scenes and States.

Scene Formula x	w	v	**Trouble**
$x_4 = \mathsf{off}$, $x_2 \ll x_1$	$x, s1$	$x, s12$	A
$x_5 = \mathsf{off}$, $x_2 \ll x_1$	$x, s1$	$x, s12$	HW
$x_3 = \mathsf{on}$, $x_2 \ll x_1$	$x, s2$	$x, s12$	CW
$x_2 \ll x_1, x_7 \ll x_1, x_6 = \mathsf{n}$	$x, s3$	$y, s2$ (where $y_6 \neq \mathsf{c}$)	ER

5. Conclusions

The main conclusions of this work are as follows:

Efficiency: The proposed solution increases the efficiency of domestic hot water heating systems using renewable energies. Normally, the thermal profitability of this type of installation, powered by solar energy, represents a saving of around 80% of that required for heating [1]. The use of the system presented in this paper represents an additional saving of $(100 - x)\%$ (depending on the position x of the lowest level sensor with respect to the bottom of the tank).

Multiplicity: The water heater itself has three different behaviors, depending on its load level. In principle, when the solar thermal input is sufficient, it works as a conventional domestic hot water storage tank powered by solar thermal energy. In a second state, when the heat transfer fluid coming from the panels has a lower temperature than the water in the tank to be heated, it cuts off the external supply and becomes atmospheric, allowing the interior water to be used without any possible mixing with the water at a lower temperature from the network. In the third state, when a minimum level is reached, it will behave similar to an electric heater, powered by renewable or non-renewable energies, working on a minimum level of water compared to the original.

Economy: The system would allow an estimated global saving of $0.80 \times \frac{100-x}{100}$ based on the conventional energy used for the heating process. In addition, it allows to use all the water contained in the tank at the maximum temperature reached (until reaching the minimum load level marked by the probe located in the lowest zone) without mixing it with the water coming from the supply network (which would cool the mixture at a considerable speed). Likewise, when the outside temperature does not allow the stored water to be heated, it considerably reduces the consumption of electrical energy by significantly reducing the volume of water that will be used from that moment onwards (it practically becomes a heating as the water flow passes through).

Complexity: The resolution by means of propositional dynamic logic has overcome the disadvantages of traditional systems to detect faults in the logical operation of the system. In addition, it contains programs that allow qualitative control of the states of the variables, with built-in qualitative addition operations, which gives the system great flexibility. This type of operation is very useful in home automation as many applications require qualitative regulation.

6. Patents

In parallel to this paper was developed the patent number ES2627209B2 [32].

Author Contributions: Data curation, F.G.; Formal analysis, A.B.; Methodology, S.M. and J.M.; Software, F.G.; Supervision, A.B.; Writing—original draft, S.M. and J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We thank the University of Malaga and CBUA (funding for open access charge: Universidad de Málaga/CBUA) for the support of this paper, and we thank also the anonymous reviewers whose suggestions helped improve and clarify this manuscript. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guzmán, F.; Merino, S. *Domótica: Gestión de la Energía y Gestión Técnica de Edificios*; RA-MA Editorial: Madrid, Spain, 2015.
2. Kawamura, H.; Naka, K.; Yonekura, N.; Yamanaka, S.; Kawamura, H.; Ohno, H.; Naito, K. Simulation of I–V characteristics of a PV module with shaded PV cells. *Sol. Energy Mater. Sol. Cells* **2003**, *75*, 613–621. [CrossRef]
3. Merino, S.; Guzmán, F.; Martínez, J. Metadomotic optimization using genetic algorithms. *Appl. Math. Comput.* **2015**, *267*, 170–178. [CrossRef]
4. Moretón, R.; Lorenzo, E.; Navarrete, L. Experimental observations on hot-spots and derived acceptance/rejection criteria. *Sol. Energy* **2015**, *118*, 28–40. [CrossRef]
5. Orozco, M.L.; Ramírez-Scarpetta, J.M.; Spagnuolo, G.; Ramos-Paja, C.A. A method for simulating large PV arrays that include reverse biased cells. *Appl. Energy* **2014**, *123*, 157–167. [CrossRef]
6. Petrone, G.; Spagnuolo, G.; Zhao, Y.; Lehman, B.; Ramos-Paja, C.A.; Orozco, M.L. Control of Photovoltaic Arrays, Dynamical Reconfiguration for Fighting Mismatched Conditions and Meeting Load Requests. *IEEE Ind. Electron. Mag.* **2015**, *9*, 62–76.
7. Orozco-Gutiérrez, M.L.; Spagnuolo, G.; Ramírez-Scarpetta, J.M.; Petrone, G.; Ramos-Paja, C.A. Optimized Configuration of Mismatched Photovoltaic Arrays. *IEEE J. Photovolt.* **2016**, *6*, 1210–1220. [CrossRef]
8. Sánchez, F.J.; Sotorrío, P.J.; Heredia, J.R.; Pérez, F.; Sidrach-de-Cardona, M. PLC-Based PV Plants Smart Monitoring System: Field Measurements and Uncertainty Estimation. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2215–2222. [CrossRef]
9. Selen, J.; Adan, I.; Kapodistria, S.; Van Leeuwaarden, J. Steady-state analysis of shortest expected delay routing. *Queueing Syst.* **2016**, *84*, 309–354. [CrossRef]
10. Quashing, V.; Hanitsch, R. Numerical simulation of current-voltage characteristics of photovoltaic systems with shaded solar cells. *Sol. Energy* **1996**, *56*, 513–520. [CrossRef]
11. Chan, C.; Coghill, G.; Liu, H. Recent advances in Fuzzy Qualitative Reasoning. *Int. J. Uncertainty Fuzziness Knowl. Based Syst.* **2011**, *19*, 417–422. [CrossRef]
12. Chavez, G.; Zerkle, D.; Key, B.; Shevitz, D. Relating confidence to measured information uncertainty in qualitative reasoning. In Proceedings of the 2011 Annual Meeting of the North American Information Processing Society, El Paso, TX, USA, 18–20 March 2011; pp. 1–6.
13. Duckham, M.; Lingham, J.; Mason, K.; Worboys, M. Qualitative reasoning about consistency in geographic information. *Inf. Sci.* **2006**, *176*, 601–627. [CrossRef]
14. Kuipers, B.; Byun, Y.T. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robot. Auton. Syst.* **1991**, *8*, 47–63. [CrossRef]
15. Liu, H.; Brown, D.; Coghill, G. Fuzzy qualitative robot kinematics. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 808–822.
16. Mossakowski, T.; Moratz, R. Qualitative reasoning about relative direction of oriented points. *Artif. Intell.* **2012**, *180*, 34–45. [CrossRef]
17. Dubois, D.; Ali, A.H.; Prade, H. Making fuzzy absolute and fuzzy relative orders of magnitude consistent. *Fuzzy Sets Syst.* **2003**, *2003*, 694–701.
18. Ali, A.H.; Dubois, D.; Prade, H. Qualitative reasoning based on fuzzy relative orders of magnitude. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 9–23. [CrossRef]
19. Agell, N.; Sánchez, M.; Prats, F. Modelos cualitativos de órdenes de magnitud: Introducción a los modelos mixtos. *Rev. Iberoam. Intel. Artif.* **2000**, *4*, 58–65. [CrossRef]
20. Dubois, D.; Prade, H. Fuzzy arithmetic in qualitative reasoning. *Model. Control. Syst.* **1989**, *121*, 457–467.
21. Travé-Massuyès, L.; Piera, N. The orders of magnitude models as qualitative algebras. *Proc. Elev. Ijcai* **1989**, *2*, 1261–1266.
22. Piera, N.; Sánchez, M.; Travé-Massuyès, L. Qualitative operators for order of magnitude calculus: Robustness and precision. In Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics, Dublin, Ireland, 22–26 July 1991; Volume 22, pp. 1–6.
23. Harel, D.; Kozen, D.; Tyurin, J. *Dynamic Logic*; MIT Press: Cambridge, MA, USA, 2000.
24. Bonner, J.; Kiffer, J. An overview of transaction logic. *Theor. Comput. Sci.* **1994**, *133*, 205–265. [CrossRef]
25. Kowalski, R.A.; Sergot, M.J. A logic–based calculus of events. *New Gener. Comput.* **1986**, *4*, 67–95. [CrossRef]
26. McCarthy, J.; Hayes, P.J. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence*; Meltzer, B., Michie, D., Eds.; Edinburgh University Press: Edinburgh, UK, 1969; Volume 4, pp. 463–502.
27. Pratt, V.R. A near-optimal method for reasoning about action. *J. Comput. Syst. Sci.* **1980**, *20*, 231–254. [CrossRef]
28. Goré, R.; Widmann, F. An optimal on-the-fly tableau-based decision procedure for PDL-satisfiability. *Lect. Notes Comput. Sci.* **2009**, *5663*, 437–452.
29. De Giacomo, G.; Massacci, F. Combining deduction and model checking into tableaux and algoritms for converse—PDL. *Inf. Comput.* **2000**, *162*, 117–137. [CrossRef]
30. Hudstadt, U.; Schmidt, R.A. A comparison of solvers for propositional dynamic logic. In Proceedings of the Second International Workshop on Practical Aspects of Automated Reasoning (PAAR-2010), Edinburgh, UK, 14 July 2012; Volume 9, pp. 63–73.

31. Lange, M. Model checking propositional dynamic logic with all extras. *J. Appl. Log.* **2006**, *4*, 39–49. [CrossRef]
32. Burrieza, A.; Guzmán, F.; Martínez, J.; Merino, S. Acumulador, Sistema y Procedimiento Para Proporcionar Agua Caliente Sanitaria. Patent ES2627209B2, 31 March 2017.