Real-time Diagnosis of Ambient Environments Using a Modeling of Physical Effects Combined with Temporal Logic

Ahmed Mohamed
SUPELEC Systems Science (E3S)
Department of Computer Science
3 rue Joliot-Curie 91192 Gif-sur-Yvette Cedex, France
+33 (0)1 69 85 14 76
ahmed.mohamed@supelec.fr

Christophe Jacquet
SUPELEC Systems Science (E3S)
Department of Computer Science
3 rue Joliot-Curie 91192 Gif-sur-Yvette Cedex, France
+33 (0)1 69 85 14 90
christophe.jacquet@supelec.fr

Yacine Bellik
LIMSI
Bât 508, Plateau du Moulon
B.P.133, 91403 Orsay Cedex France
+33 (0)1 69 85 81 10
yacine.bellik@limsi.fr

Abstract — **Ambient intelligence systems interact with their surroundings using actuators and based on environmental data collected from sensors' readings. Diagnosis in this context must address some particular challenges due to the dynamic nature of these systems and the impossibility to pre-define control loops between sensors and actuators at design time. A possible solution to this problem is to base diagnosis on observed physical phenomena (effects) induced by actuators and to reason over a pre-defined ontology allowing one to apply physical laws, to compare calculated values with actual sensors' readings and thus to notice anomalies which corresponds to probable faults. This "effect"-based model, which describes the expected physical effects of the actuators onto the environment, allows one to perform basic diagnosis, using a static view of the system. However, to perform more complete diagnosis, we claim that one has to take the dynamics of the system into account. To achieve this, this paper proposes to extend the simple "effect"-based model with a behavioral model using temporal logic.**

Keywords-Ambient intelligence; ubiquitous systems; sensor; actuator; diagnosis; OWL; ontology; reasoning; physical law; temporal logic

I. INTRODUCTION

Ambient intelligence systems are interactive systems that have an overall goal of satisfying users' needs in everyday life tasks using the least intrusive way. Such systems interact with their environments using actuators and sensors. The data collected by the latter keep the system aware of its environment. Depending on the task intended, the system uses these data to determine the actions to take using the necessary actuators in order to achieve the current task. In this context, the system must have the means to check autonomously whether the actions are performed correctly. As a matter of fact, when the ambient system sends out orders to an actuator, the information provided in return from the latter reflects only the receipt state of the transmitted orders, not their actual execution. For instance, when the system activates a light bulb, it does not know if the light has really been switched on (for instance due to a damage to the bulb itself).

The particularity of ambient systems is that, unlike traditional systems, physical resources (mainly sensors and actuators) are not necessarily known at design time. In fact they are dynamically discovered and may appear and/or disappear at run-time (depending for instance on user location), so control loops cannot be pre-determined. That is why control theory that is usually used to pre-determine closed control loops using ad-hoc sensors is not applicable to this type of highly dynamic systems. The model proposed in [1] is a framework for building dynamically the equivalent of control loops for ambient systems, by using available resources at a given time and using them to perform *diagnosis* at run-time. The approach is based on the modeling of the physical phenomena (so-called *effects*) expected in the environment and that may be produced by actuators and detected by sensors. This method has proven itself to be well adapted to the dynamic nature of ambient systems, since it enables the system to automatically associate actuators and sensors, and thus, to deduce the expected measurement provided by a given sensor when a certain action is performed by an actuator (for instance, an increased light level may be expected when a bulb is activated). This way, the system is able to produce an accurate diagnosis at run-time while allowing one to totally decouple actuators and sensors at design time. However, deducing faults in such a situation might depend on the previous state of the system and of the environment (for instance, an error consisting in an unexpected drop in light level is detected by comparing the current light level with the previous one), thus it is crucial to consider their overall temporal behavior. For this reason, this paper introduces temporal extensions to the diagnosis framework proposed in [1].

The remainder of this paper is organized as follows. Section 2 exposes the architecture of the diagnosis framework and shows the required extensions so as time constraints can be taken into account. Then, Section 3 presents a complete example demonstrating our approach.

 ISBN: 978-1-61208-171-7

Finally, the conclusion highlights some issues for future work.

II. STATE OF THE ART

One of the main particularities of ambient environments is that services, which goal is generally to satisfy user's preferences by performing a specific task (for example regulating room temperature) or assisting him/her in his/her task (like assisting a user in some kitchen tasks), are executed in the background in a way that they are unnoticeable by the user. Diagnosis in ambient environments can correspond to either verifying that the user has properly done his expected task, in which case it is a user-behavior diagnosis, or verifying whether the system actuators have performed their task properly, in which case it is system-behavior diagnosis. This requirement, building non intrusive ambient systems, causes some difficulties in fault detection. Indeed, it is unacceptable for a non intrusive system to flood the user with a large number of fault detection data. In the same time, not informing the user of detected faults may cause that users continue to rely on failed services without noticing. So, in general, this characteristic, which is working correctly in the background, shows how crucial the diagnosis task is. Moreover, ambient systems are becoming increasingly autonomous and complex, which makes diagnosis a nontrivial task [2].

Many techniques are proposed for fault detection, for instance in some assisted living systems (called also smart homes); the approach consists in gathering user data (behavior, preferences, etc.) in order to apply machine learning techniques [3] to detect anomalies in user behavior. This approach allows us to perform user-behavior diagnosis. With our work, what we are aiming for is a real-time system-behavior diagnosis framework (by device we mean actuators and sensors). In fact complex systems fault detection techniques can be used in the case of device-centered diagnosis. The challenge here is to consider the most suited approaches to ambient systems' characteristics and to adapt them if possible. One of these approaches proposed for complex systems diagnosis is the model-based diagnosis technique. It is a technique based on a system description that is used to define the behavior of each component within the system and the connections between these components [4]. The technique consists in simulating the system's behavior and reasoning over the system model. Obtained information is used to compare the expected system behavior with the actual system behavior, and thus to detect faults. The major challenge of this technique is combinatorial explosion which makes the approach useless for devices composed of a considerable number of components [5].

In general, we notice that regardless of the approaches proposed in existing work, it is always supposed that sensors and actuators, represented in the model, are somehow directly linked. In other words the model explicitly contains the relationships that link actuator actions and sensor states. We claim that building such explicit links is poorly adapted to highly dynamic ambient systems. Indeed, as devices are added to and removed from an ambient environment at runtime, it is very difficult for the system designer to thoroughly describe the system at design time. For these reasons, we introduce our approach allowing the decoupling of actuators and sensors in the model, while enabling the system to deduce the links between them at runtime.

III. THE DIAGNOSIS FRAMEWORK

Before explaining the effect-based model and the behavior of the diagnosis process, let us introduce the context of use of the diagnosis framework. In Fig. 1, the diagnosis framework is situated within the context of an ambient system and its main components are illustrated. It is composed of an effect meta-model and a diagnosis process. The effect meta-model is instantiated to reflect the static representation of the ambient system (static model); it contains the actual system components along with the expected physical phenomena to be observed in the environment. The dynamic model defines the dynamic behavior of possibly complex physical phenomena. The union of the dynamic and static model constitutes the "system model instance". The so-called "diagnosis process" performs run-time, background diagnosis on the ambient system, based upon information drawn from the system model instance and the ambient system itself. As illustrated by the directions of the arrows going toward the ambient system from the diagnosis framework, the latter is designed in such a way that it may be "grafted" onto the ambient system without changing it.

It is to be noted that in this paper we do neither discuss the modeling, nor the operation of the ambient system. We do rather discuss, in the following subsections, the modeling and the use of the effect meta-model, its possible instances and the diagnosis process.

A. *The Effect Meta-Model*

1) *The Static Model*

In order to have a generic approach we propose a meta-model that is based on the modeling of ambient objects (mainly actuators and sensors) and the explicit description of the concept of *effect*. The latter becomes the only "deduced (via reasoning)" link between actuators and sensors. This meta-model is instantiated to represent the diagnosed ambient system. To benefit from good extensibility properties and broad tool support for later software implementation of the diagnosis framework, ontologies, namely OWL ontologies [6], have been used to design the effect-based meta-model.

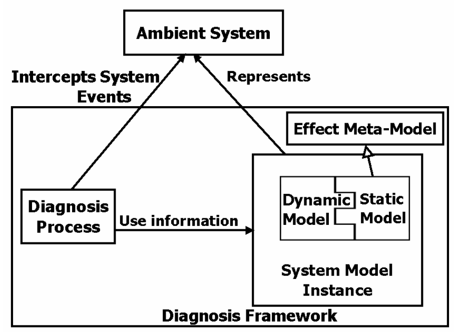

Figure 1. The Diagnosis Framework and the Ambient System

In the proposed approach, illustrated in Fig. 2 by the structure of the effect meta-model ontology, the concept of effect defines the relation between actuators and sensors. This definition is done in respect of the description of the physical consequences of the actuators' actions on the ambient environment and thus on the sensors' readings. Such design requires an explicit definition of the physical law. However this definition of physical laws is more or less detailed so the model (instance of the meta-model representing the actual ambient environment on which diagnosis is performed) can follow different levels of granularity. The choice of the latter can depend, among other things, on the context of use, for instance assisted living homes for blind persons would have a detailed definition of the model for the propagation of sound waves.

The main contribution of this approach, as illustrated by Fig. 2, is to eliminate any direct link, at design time, between sensors and actuators in an ambient environment. For example in an environment composed of a light bulb (actuator) and a light sensor (sensor), the light bulb emits (produces) light (effect). Light is characterized by light intensity (effect property). Light sensor is sensible to (detects) its surrounding light intensity (measurable property). To calculate (calculates) the light intensity (measurable property) that reaches the light sensor from the light bulb considering the distance between them, we model the fact that light intensity decreases with the square of the distance [7] (physical law). In the mathematical formula of this physical law the distance between the light bulb and the light sensor must be expressed. The distance between the two components is deduced from their respective positions (ambient object property). Once we have the results of the calculations of the physical law which is the light intensity we expect around the light sensor, and we have the current value of the light intensity given by the sensor itself, the diagnosis is performed by comparing, according to some diagnosis strategy, the two values. With this model we do not impose a diagnosis strategy. So in general all the information provided by the model is in fact the measurable physical properties values that are calculated by the corresponding physical laws. These are the values that are expected to be read by the sensors. These values are then compared with their equivalent measurable physical properties values that are given by the sensors' readings.

As stated earlier, the physical laws can follow different levels of details. The benefits of such dynamicity can be demonstrated when considering different contexts of use. Let us consider the lighting system as an example. Let us say we are in the context of an ambient home lighting system; in an ambient home we can imagine a light propagation formula as a simple ON/OFF relation between light bulbs emitted light and light sensors' readings. However lighting a work space might use more fine-grained rules, so in this context the formula would use a more accurate light propagation law (like the previously mentioned inverse square law) to make sure that light intensity remains around the expected value. It is up to the final designer of the actual ambient system to determine the level of granularity appropriate to the context.

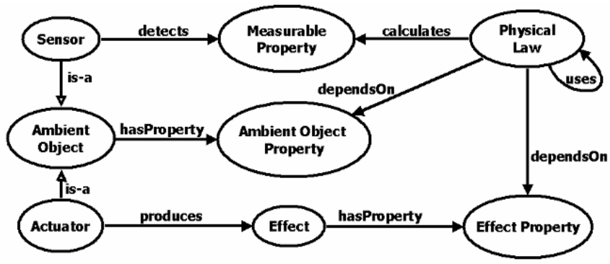

Figure 2. The effect meta-model ontology schema

The main goal of this approach is to provide a dynamic diagnosis framework. The effect meta-model provides this diagnosis framework with the necessary data. This data is used by the diagnosis process to perform diagnosis.

2) The Dynamic Model

The effect based meta-model models effects as physical phenomena. Frequently, the latter depends on time variables. To model temporal behavior a first solution would be to use Linear Temporal Logic (LTL). As a matter of fact in addition of being a formalism for the specification and verification of concurrent and reactive systems, LTL is in fact a formalism for expressing qualitative properties about the execution of the system [8]. However when examining the behavior of the actuators in an ambient environment, it is noticeable that, from the time actuators are activated, most of the times, the physical impact takes a certain delay before it is observed. The durations of these delays vary depending on the type of the physical phenomena. For instance after turning on a heater, the heat effect that is supposed to be produced is not noticeable until a certain time has passed, the length of this time is defined by heat transfer laws. Such properties cannot be taken into account by using classical linear-time temporal logic (LTL). For real-time systems where a run of a system is modeled as a sequence of events that are time-stamped with real values, which is the case here with times and durations calculated by physical formula, LTL is inadequate. Instead, for such systems, modalities decorated with quantitative constraints over real values are required. A known extension for such logic is MTL (Metric Temporal Logic) in which modalities of LTL are enriched with quantitative constraints [9]. With MTL when describing the behavior of real-time system one can consider deadlines between environment events and corresponding system responses. For example "*every "alarm" is followed by a "shutdown" event in 10 time units unless "all clear" is sounded first*" [10] can be represented as:

$$\square(alarm \rightarrow (\lozenge_{(0,10)} allclear \; \square \; \lozenge_{\{10\}} shutdown))$$

$\lozenge_{(0,10)}$ means sometime in the next 10 time units.

$\lozenge_{\{10\}}$ means in exactly 10 time units.

Although there are other alternative approaches to extend LTL such as Timed Propositional Temporal Logic (TPTL) [11], MTL meets our needs at this stage.

B. The Diagnosis Process

The diagnosis process is a set of finite state machines modeling the system's behavior. It is using sensors and actuators related events as transitions of the ambient system

 ISBN: 978-1-61208-171-7

behavioral model to perform diagnosis tasks, hence the relation “Intercepts System Events” between the diagnosis process and the ambient system in Fig. 1. In fact the diagnosis process is a generic process that performs diagnosis based on one hand, the ambient system’s behavioral model and, on the other hand, information from the system effect model (instance of the effect meta-model). For example we can imagine a light diagnosis task consisting in expecting an increase of the light intensity value after light is turned on, or we can imagine a continuous light intensity verification diagnosis process, during which the diagnosis task consists in verifying that light intensity value is kept around a certain value. The latter value changes according to both the received system event (light turned OFF or ON) and/or information deduced from the instance of the effect meta-model (light intensity value deduced from the distance between light sources and light sensors).

1) The concept of time in the diagnosis framework

In the proposed approach, the issue of “time” is considered from two angles; the first angle is time as a physical variable in the physical formulas, the second angle is time as part of the diagnosis framework dynamics (behavioral model). In the first angle, time is used in the physical formulas defined in the Static Model (instance of the effect meta-model). The fact that time is a shared concept between the Static and the Dynamic model is the reason that the system model instance is divided into two interrelated parts as illustrated in Fig. 1. When present in these formulas, time becomes a shared concept and, thus, the relation between the Static Model and the Dynamic Model. The latter, if necessary, uses time in the description of the physical phenomena’s behavior, in which case is represented as a behavioral model. As for the diagnosis process, it describes the system’s behavior while taking into account the physical phenomena’s impact on the system’s overall behavior, which requires interacting with the Dynamic Model’s behavioral model; this is the second angle in which time is considered. The diagnosis process intercepts ambient system events to perform diagnosis (the technique is detailed in the next subsection). The challenge here is to consider both angles and their combination into one diagnosis dynamic framework capable of performing real time fault diagnosis. What is to be dealt with here also is the synchronization of time value with actual system’s time. It is the diagnosis process part of the framework that handles this task.

IV. A DIAGNOSIS EXAMPLE

In this example, we will see how diagnosis is performed when a bathtub is being filled. As illustrated in Fig. 3, we have a bathtub and four actuators controlled by the system’s controller: two water taps (a hot one and a cold one), a water drain, and a resistor. The later role will be explained in the second part of the example. There are also two sensors: a thermometer and a level indicator, whose readings keep the system informed about the state of the environment (water temperature and level) in real-time.

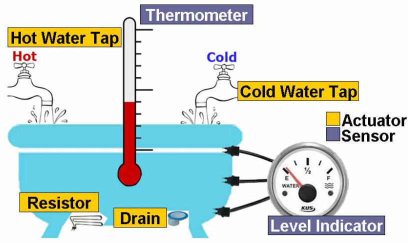

Figure 3. Components of the Bathtub Diagnosis Example

We suppose that the provided ambient system’s behavioral model is composed of a set of finite state machines (FSM) describing the system’s overall behavior. In this example, we isolate the part that describes tasks that are related to the bathtub behavior. Fig. 4, is a simplified proposal of what the bathtub FSM would be. In this demonstrative example, we will see a simplified diagnosis example on a specific task; corresponding to the “filling bathtub” state of the FSM. The latter task and its relative transitions are the parts that are bold in Fig. 4.

For this particular example, the temperature value that is requested by the system is 50°C and the level is 150 liter. This is represented by the entering transition to “filling bathtub”, the instantiation of this transition is:

Start Filling [50 ; 150]

The diagnosis process part of the diagnosis framework as illustrated in Fig. 1, listens to system events (Start Filling [50 ; 150]). Afterward, the diagnosis process would start performing diagnosis tasks related to the “filling bathtub” state. In this example, we will consider a simple diagnosis task consisting in comparing, at every point in time, the expected values of water level and temperature with actual values read by the sensors. The comparison takes into account a tolerance value defined by the diagnosis process as a parameter of the physical law instance. A global variable “time” is set to keep track of time elapsed since the beginning of the diagnosis process. The “time” unit is chosen to be “seconds” so no conversion is needed when used in physical laws. Physical laws, associated to both water temperature and water level, involve quantitative time constraints that can be described using MTL.

In the first part of this example, only physical laws that are related to water level are considered. The diagnosis of water temperature is dealt with in the second part. The mathematical formulas of these physical laws are:

Water Flow Ambient Law:

$$\begin{gathered}\texttt{Ambient\_Water\_Quantity=}\\ \texttt{Water\_Quantity(Hot)+Water\_Quantity(Cold)}\end{gathered} \tag{1}$$

Water Flow Law for Hot and Cold Water:

$$\begin{gathered}\texttt{Water\_Quantity(Hot)=}\\ \texttt{L0(Hot)+Water\_Discharge\_Rate(Hot)xtime}\end{gathered} \tag{2}$$

$$\begin{gathered}\texttt{Water\_Quantity(Cold)=}\\ \texttt{L0(Cold)+Water\_Discharge\_Rate(Cold)xtime}\end{gathered} \tag{3}$$

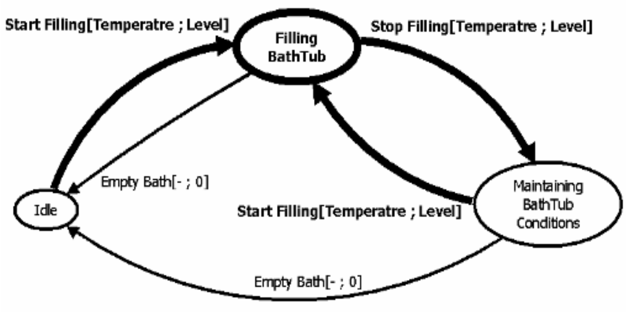

Figure 4. Simplified FSM describing bathtub behavior

It is to be noted that "Ambient Water Level", which is a sensor's reading given by the water level indicator in liters, and "Ambient Water Quantity" which is calculated by the *Water Flow Ambient Law* in cm^3, represent the same entity, which means that they are comparable entities after applying a simple rule of physical unit conversion from liter to cm^3. Moreover, L_0 (Initial level) is considered as null for simplification reasons. These physical laws and other components of the effect meta-model instantiating the ambient system by the diagnosis process are represented by the rectangles in Fig. 5.

The diagnosis process performing bathtub water level diagnosis uses information taken from this instance of the effect-based meta-model corresponding to every point of time diagnosis is performed. So, knowing the system's water discharge rate value for hot and cold water, at any given time (timer value) the diagnosis process knows both the value of the water level detected by the level indicator sensor and the value of the expected water level calculated by the stated physical laws. This information is used to perform diagnosis. Let us suppose that we have a constant "Water Discharge Rate" of $140cm^3/s$ for Cold water and $110cm^3/s$ for Hot Water. Let us also suppose that diagnosis over water level is performed periodically every 3 seconds. TABLE I illustrates the trace of the diagnosis process for the first 15 seconds after the order to the actuators (water taps) has been transmitted ("timer"=0 being the moment the order has been transmitted).

The first two null values given by the level indicator sensor at the first and second diagnosis can be explained by the fact that 750 cm^3 of water is not enough to fill the bathtub floor so that water is detected by the sensor that is fixed usually on the bathtub side. In this example, we insist on the fact that, so far, the output of the diagnosis are information describing the expected state of the system after the proper execution of the system's command and that the framework does not impose a way of using the generated diagnosis information, nor how to compare them with actual sensors' readings. The diagnosis results might be used for textual warnings to the user of the ambient system as a feedback on what is going on and whether or not its requested actions are being properly executed by the system, or, in other cases, it might be used by the ambient system itself as input information to a certain control mechanism for fault correction.

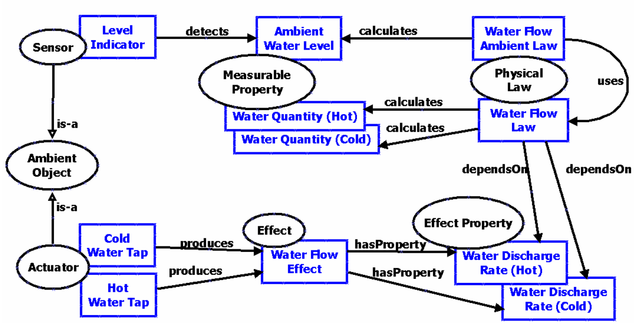

Figure 5. Effect-based model instance implementing the static model related to bathtub level diagnosis

TABLE I. WATER LEVEL DIAGNOSIS TRACE FOR THE FIRST 15 SECONDS

Time	Ambient Water Quantity *(From effect Model)*	Ambient Water Level *(From Level Indicator)*	Diag-nosis
0 s	0.00 cm^3 (0.00 liter)	0.00 liter (± 2)	OK
3 s	750.00 cm^3 (0.75 liter)	0.00 liter (± 2)	OK
6 s	1500.00 cm^3 (1.50 liter)	1.42 liter (± 2)	OK
9 s	2250.00 cm^3 (2.25 liter)	2.00 liter (± 2)	OK
12 s	3000.00cm^3 (3.00 liter)	2.67 liter (± 2)	OK
15 s	3750.00 cm^3 (3.75 liter)	3.04 liter (± 2)	OK

These control mechanisms have the particularity to be created at run-time. Using available actuators, those control mechanisms would have been used to correct water level when a fault is reported by the diagnosis process. For instance this can be done by increasing the water discharge rate when the level is less than expected and opening the water drain when the level is more than expected. The issue of system's behavioral control is not detailed in this paper.

It is to be noted that when dealing with water level diagnosis the dynamic part of the system model instance is not involved since we consider that in this case there are no non-negligible physically defined delays between actuator actions (filling bathtub with water) and the sensors responses (detecting the corresponding water level in the bathtub). This is, of course, not the case in the second part of the example which is the water temperature diagnosis part.

In this second part of the example, we consider that the bathtub offers a "hot tub" functionality. Water already present in the bathtub is heated by an immersed heating element that is basically composed of a resistor that converts electric power into heat. This heating element will be referred to as "resistor" for the rest of the paper. In this particular case we suppose that our bath tub electric heating system has a power rating of 2kW. We also suppose that water comes only from the cold water tap. What we notice here is that the water temperature elevation is incremental over time. In fact the time between the moment in which the heating element starts heating the water to a certain temperature and the moment in which the water reaches that temperature is non-negligible. Thus, this delay in detecting,

 ISBN: 978-1-61208-171-7

by the thermometer (sensors in general), the heating action (the physical phenomena's actions) done by the resistor (actuators) on the water should be taken into consideration and should appear somewhere in the system model. In reality, from physics point of view, the incremental heat elevation is caused, according to enthalpy theory [12], by total accumulated quantity of energy Q added to the system by the actuator, this value is calculated using an integration of the instantaneous amount of power P generated by the resistor (we will call this effect "Heat Emission Effect") over time:

$$Q = \int_{[ti,tf]} P(t)dt \text{ [joule]} \quad (4)$$

where t_i is the instant where the effect starts and t_f is the instant where the effect ends. To be able to perform discrete calculations, this integral is converted into a sum of instant power values in time:

$$Q = \sum_{[ti,tf]} P(t) \text{ [joule]} \quad (5)$$

It is to be noted that in this method the calculated current temperature value depends on both, the current produced power value (which is generated by the resistor), and the previous (at t-1) calculated energy value. To calculate the ambient temperature of the water we use the enthalpy formula that states that at a constant volume and pressure:

$$v.c=\Delta H/\Delta T \quad (6)$$

where c is the water specific heat capacity, which is the amount of heat required to change water's temperature (The volumetric heat capacity of water is 4.1796 $J.cm^{-3}.K^{-1}$ [13], conversions from Kelvin to Celsius ought to be considered later), v is the total volume of the water (its value in cm^3 can be deduced at any given time using (1), (2) and (3)), ΔH is the enthalpy variation and ΔT is the temperature variation. Under constant (atmospheric) pressure the quantity of heat Q received by a system is equal to its enthalpy change ΔH. So a body of volume v where the temperature (which is the value to be calculated and compared with thermometer reading) varies from t_i to t_f receives the amount of heat:

$$Q = \Delta H \quad (7)$$

To apply this to the effect-based model, an effect representing the heat emission from the actuator "resistor" is instantiated. We call it "heat emission effect"; this effect has the property power that we will call "heat power" (the instantaneous amount of power P described earlier). The later along with other properties related to other effects, other actuators and/or other sensors will be used to evaluate all the previously stated laws that are related to "heat emission effect". Indeed, results from water level physical laws (1), (2) and (3) are to be used in heat related laws. As for the values, to remain consistent with previous results for water level diagnosis, we consider now that the cold water discharge rate is $250cm^3/s$ (which was previously the sum of hot and cold water discharge rate), and that the hot water tap is closed. With this configuration we obtain the same results for water level diagnosis as the first part of the example. We also consider that we have the property "heat power" (with the value of 2500J/s) as an effect property of the "heat emission effect" produced by the actuator "resistor". We also suppose that we have a constant loss of heat caused by the direct contact of the water with ambient air and the bathtub material, this heat loss is represented by a "heat power" of -500J/s; to differentiate from previous heat property we call this property "heat loss". The model is flexible in the sense that it offers many ways to represent this loss in heat; the only constraints are to have an effect property of type "heat power" and of a negative value. So to align this idea to the effect model, the "bathtub" itself is instantiated as an actuator so that it can produce "heat emission effect" with "heat power" value of -500J/s. As a total we then have a total "heat power" of 2000J/s produced by the combination of heat loss and the resistor. The resulting instances in the effect model are illustrated in Fig. 6, in which, the 4 heat related physical laws are simplified to one instance.

During the first 3 minutes (180 seconds), we obtain the temperature diagnosis traces illustrated in TABLE II (traces are taken every 30 seconds and initial temperature variation is considered to be null "0K").

To better understand the results let us consider the diagnosis at the second 150.

- The water quantity calculated by the Water Flow Ambient Law is $37500cm^3$ [$=250cm^3.s^{-1}$x150s].
- The accumulated water heat energy, calculated by (5), is 300000J [$=2000J.s^{-1}$x150s].
- The ambient water temperature is calculated by (6) and (7) as it is the result of the temperature augmentation at t=150s, which is 1.9140K [=300000/(v.c); where $v=37500cm^3$; and c=4.1796 $J.cm^{-3}.K^{-1}$], plus the temperature at t=149s, which is equal to 285.1947K. The final result is 287.1088K (13.95°C).

The latter value is compared with the sensor reading which is 13.07°C, the comparison gives a successful diagnosis since we have a tolerance margin of 2°C.

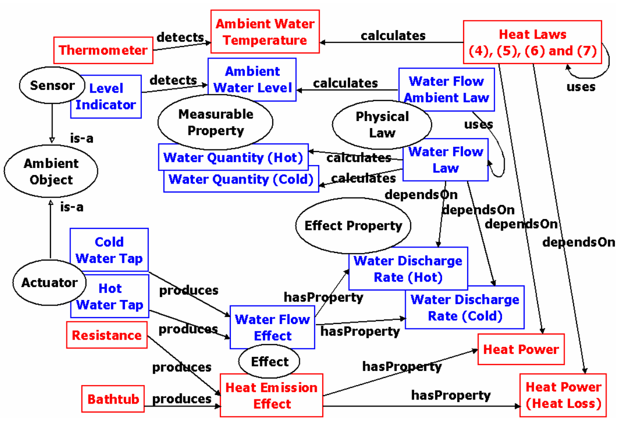

Figure 6. Effect-based model instance implementing the water temperature diagnosis.

TABLE II. WATER TEMPERATURE DIAGNOSIS TRACE

Time (s)	Water Quantity "Calculated by (1), (2) and (3)" (cm^3)	Accumulated Water Heat Quantity "Calculated by (5)" (joule)	Ambient Water Temperature "Calculated by (6) and (7)" (K)	Ambient Water Temperature "From Thermometer Reading" (°C)	Diagnosis
0	0	.	.	17.03(± 2)	Fault
30	7500	60000	57.42 (-215.72°C)	15.79(± 2)	Fault
60	15000	120000	114.84 (-158.30°C)	13.23(± 2)	Fault
90	22500	180000	172.26 (-100.88°C)	11.64(± 2)	Fault
120	30000	240000	229.68 (-43.46°C)	10.02(± 2)	Fault
150	37500	300000	287.10 (13.95°C)	13.07(± 2)	OK
180	45000	360000	344.53 (71.38°C)	69.09(± 2)	OK

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced an original method for the diagnosis of ambient systems; the method is based on a diagnosis framework. This framework is composed of a diagnosis process and an effect-based model. The effect-based model takes into account the particularities of ambient environments (no predetermined relation between actuators and sensors). We introduced an effect-based model to identify the links between actuators and sensors depending on the physical effect produced by the actuators and the physical properties detected by sensors, the links are defined by the corresponding physical laws. In addition of its compatibility with ambient systems, this method offers the freedom to choose the level of detail in which the system is described depending on the context of use, since the physical laws can follow different levels of granularity. Along with the effect-based model the system model is composed of a dynamic model that describes some of the physical phenomena's behavior and a diagnosis process that uses the information from the other models to perform real-time diagnosis.

As future work we envision to fully evaluate the diagnosis process part of the model and the dynamic model part of the framework. The current framework is designed mainly for fault detection (discovering the existence of fault) is not handled yet. We consider adding a probabilistic model for error isolation. The idea is to label the devices with a failure probability value, so when an error is detected, we would have additional information for the identification of its source. Although the user is the center of an ambient intelligent system, as the main purpose of the system is to satisfy his/her preferences, the user is not yet represented in our proposed model. In fact, contrary to the ambient systems' behavior which is on many levels predictable and thus can be modeled, the behavior of users is unpredictable, which makes its modeling intricate. However explicitly modeling user behavior, tasks and needs would allow the diagnosis framework to perform more accurate diagnosis. Finally, real-scale tests in an experimental ambient environment will be carried out in order to validate the diagnosis framework.

ACKNOWLEDGEMENT

This work has been performed within the CBDP project, a project co-funded by the European Union. Europe is involved in Région Île-de-France with the European Regional Development Fund.

RREFERENCES

[1] A. Mohamed, C. Jacquet, and Y. Bellik, "Diagnosis of Ambient Systems Based on the Modeling of effects", The International Conference on Ambient Systems, Networks and Technologies. Paris, 2010, pp.35-44.

[2] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the Physical World with Pervasive Networks" , IEEE Pervasive Computing, January-March 2002, pp.59-69.

[3] JC. Augusto, P. McCullagh, V. McClelland, and J-A. Walkden. "Enhanced Healthcare Provision through Assisted Decision-Making in a Smart Home Environment", proceedings of the 2nd Workshop on Artificial Intelligence Techniques for Ambient Intelligence, 2007.

[4] Kitts, C., "Managing Space System Anomalies Using First Principles Reasoning." IEEE Robotics and Automation Magazine, Special Issue on Automation Science, v 13, n 4, December 2006, pp. 39-50.

[5] J.D. Kleer, "Focusing on Probable Diagnoses", in Proc. AAAI, 1991, pp.842-848.

[6] M. Dean and G. Schreiber, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/, latest version available at http://www.w3.org/TR/owl-ref/

[7] SP. Parker, "McGraw-Hill Dictionary of Scientific and Technical Terms", McGraw-Hill Science & Technology Dictionary, McGraw-Hill, 2003.

[8] A. Pnueli, "The temporal logic of programs". FOCS 1977. IEEE Computer Society Press, Los Alamitos 1977, pp.46-57.

[9] R. Koymans, "Specifying real-time properties with metric temporal logic", Real-time Systems 2(4) Kluwer 1990, pp.255-299.

[10] J. Ouaknine and J. Worrell, "Some Recent Results in Metric Temporal Logic", in Proc. FORMATS, 2008, pp.1-13.

[11] R. Alur and T.A. Henzinger, "A really temporal logic", Journal of the ACM 41, 1994, pp.181-203.

[12] G.J. Van Wylen and R.E. Sonntag, "Fundamentals of Classical Thermodynamics", 3rd ed. New York: Wiley, 1986.

[13] K. J. Laidler, "The World of Physical Chemistry", Oxford University Press, Oxford, 1993.