

Automatic generation of model for building energy management

Quoc-Dung Ngo
People's security academy
Hanoi, Viet Nam

Yanis Hadj-Said
G-SCOP lab, CNRS UMR 5272
Grenoble Institute of Technology, France

Stéphane Ploix
G-SCOP lab, CNRS UMR 5272
Grenoble Institute of Technology, France

Ujjwal Maulik
Jadavpur University, Kolkata-700032
West Bengal, India

Abstract—This paper proposes a model transformation approach for model-based energy management in buildings. Indeed, energy management is a large area that covers a wide range of applications such as simulation, mixed integer linear programming optimization, simulated annealing optimization, model parameter estimation, diagnostic analysis, . . . Each application requires a model but in a specific formalism with specific additional information. Up to now, application models are rewritten for each application. In building energy management, because the optimization problems may be dynamically generated, model transformation should be done dynamically, depending on the problem to solve. For this purpose, a model driven engineering approach combined with the use of a computer algebra system is proposed. This paper presents the core specifications of the transformation of a so-called high level pivot model into application specific models. As an example, transformations of a pivot model into both an acausal linear model for mixed integer linear programming optimization and a causal non-linear model for simulated annealing optimization are presented. These models are used for energy management of a smart building platform named Monitoring and Habitat Intelligent located at PREDIS/ENSE3 in Grenoble, France.

Keywords—building energy management system, model transformation, model driven engineering, optimization, mixed integer linear programming, simulated annealing

I. INTRODUCTION

Nowadays, the building sector represents about 38% of the total energy consumption in Europe and 63% in France [1, 10]. Therefore, energy consumption reduction in building has become an important challenge for researchers. A lot of Building Energy Management Systems (BEMS) have been proposed aiming at minimizing the daily energy consumption while maintaining a satisfactory level of comfort for occupants using models of the building systems. Modern building systems may be complex in terms of number of appliances, including production and storage means but also in terms of applications, which may cover functionalities like monitoring and state estimation, model parameter estimation, simulation synchronized with measurements for replays but also model based energy management using optimization algorithms, . . . Therefore, tools to handle and transform models are required. In the last decade, sophisticated methods, formalism and tools have been developed for different applications in order to better master dwelling energy consumption and production such as:

- global optimization for anticipative energy management [3, 7, 11, 12]. Actually, a day ahead energy management plan proposes to occupants the best configurations for building envelope and appliances for the next 24 hours in order to optimize a cost/comfort compromise. Optimization problem is dynamically generated according to the appliances and occupant activities impacting the management time horizon. To deal with thousands of variables and constraints in a acceptable computation time, a *mixed integer linear programming* (MILP) solver is used. Therefore, an acausal linear problem is required for this application.
- fast optimization for interactions with occupants [43]. In residential sector, building energy management cannot be fully automatized: it results from an interactive process where occupants shape the anticipative plan by modifying or adding constraints. It is often not necessary to re-perform a global optimization: a local optimization is often sufficient and more interesting because it is less time consuming when using the global solution as initialization.
- simulation for analyzing impacts of actions [46, 47]. Simulation approach can be causal such as with Matlab or acausal with Modelica. There is no optimization process but depending on the kind of simulation, the nature of the required models may differ.
- parameter estimation to learn the building intrinsic behavior thanks to recorded datasets. [44, 45]. The variables that were previously considered as parameters become yet the optimization variables while other variables are set to the values belonging to datasets.

These applications require each a dedicated formalism, whose nature may be very specific. Generally speaking, this problem is not a new one but in building energy management, where models are dynamically generated depending on the used appliances and on the question the energy manager has to solved, it cannot be handled manually by an expert.

Automatic model generation is a promising approach to avoid the above issue. gPROMS [8] or General Algebraic Modeling System (GAMS) [9] have been developed for the user to focus on the modeling problem by forgetting the application formalism requirements. Once the core model is defined, these systems manage the time-consuming transformation required for most common optimization solvers (GLPK,

CPLEX, GUROBI, . . .). Although these approaches are based on a superset high level language, they are not able to change the nature of a model: a causal model cannot become acausal, a nonlinear model cannot be linearized, a causal model for simulation cannot be transformed into a causal model for parameter estimation. To handle such transformations, models have to be deeply modified using a computer algebra system to reformulate and transform constraints.

Because the model construction of a whole dwelling is not a trivial task, due to the system complexity and dynamicity, it is not a good solution to build a whole dwelling model at once but it is preferable to compose step by step element models before generating sub-systems and finally the overall system. When there is a change, it is just needed to add or remove some element models.

In order to fit the building energy management system (BEMS) needs for automatic transformation between application models, a Model Driven Engineering (MDE) [4] approach combined with a computer algebra system (CAS) is proposed. The MDE main objective is to reduce software production cost by using standardized models and increasing their flexibility to deal with computer technology evolution. This methodology is largely implemented in object oriented modeling. This paper proposes to adapt this approach to the transformation of composed pivot model into application specific models.

The paper is composed of 5 main sections. The next section aims at formulating different key concepts using an illustrative example. The third section presents the transformation process principles. An application to the *PREDIS Monitoring and Habitat Intelligent* platform is presented in the fourth section and the last section is dedicated to analysis of two model transformation results dedicated to model based energy management.

II. PROBLEM STATEMENT

In this section, different key concepts aiming at composing an overall dwelling model based on Model Driven Engineering approach are presented. An example is used as an illustration.

A. Concept of model transformation

Let's consider resistor modeled by: $C_0 : U_1 = R_1 \times I_1$.

This simple model may be used by a designer into different optimization problem, adding information like lower and upper bounds of the possible value domains of variables or an objective function. Consider for instance the following (unrealistic) optimization problem:

$$\begin{aligned} C_0 & : U_1 = R_1 \times I_1 \\ C_1 & : R_1 \in [0, 5] \\ C_2 & : U_1 \in [0, 4] \\ \text{Objective} & : \max_{R_1, U_1} I_1 \end{aligned}$$

In spite of its simplicity, if another resistor R_2 is added in parallel, the whole system model has to be rewritten:

$$\begin{aligned} C_0 & : U_1 = R_1 \times I_1 \\ C_1 & : U_2 = R_2 \times I_2 \\ C_2 & : I_{total} = I_1 + I_2 \\ C_3 & : R_1 \in [0, 5] \\ C_4 & : R_2 \in [0, 3] \\ C_5 & : U_1 \in [0, 4] \\ C_6 & : U_2 \in [0, 4] \\ C_7 & : U_1 = U_2 \end{aligned}$$

$$\text{Objective} : \max_{R_1, U_1, R_2, U_2} I_{total}$$

Although the rewriting process is not that time consuming in this example, it becomes a tough work for complex systems which contain hundreds of variables and constraints. In addition, model may also have to be rewritten depending on the target application. For instance, some optimization algorithms require a causal ordering (Simulated Annealing, for example), some others require linearization (Mixed Integer Linear Programming, for example). Therefore, two difficulties must be dealt with:

- a model must be composed of model elements that can be reused
- a model has to be transformable.

To handle model transformation in optimization problems, the concept of pivot model is introduced. Actually, a pivot model is a high level application-independent description that can be transformed into target application formalisms, which may require a reformulation of some constraints including equations i.e. equality constraints.

In the computer science literature, model rewriting processes are usually managed using the concepts of Model Driven Engineering.

B. Concept of Model Driven Engineering

Basically, the Model Driven Engineering approach aims at separating models based on company know-how and those related to software implementations in order to maintain the sustainability of the company know-how in spite of the changes of development environment [4]. To do this, it is necessary to firstly define Platform Independent Models (PIM) i.e. pivot model, technically independent from any execution platform. It enables the generation of a set of Platform Specific Models (PSM) afterwards. Based on the MDE approach, the problem can be decomposed into 2 abstraction levels. The two concepts of PSM and PIM are corresponding respectively to the level M0 and M1. Shortly, the signification of each level is:

- level M0 (PSM) is the real system that contains executable object
- level M1 (PIM) is the model that represents the system

The main objective of MDE is to perform transformations between PIM and different PSMs. There are two types of transformations of models:

- transformation model-to-code (PIM to PSM)
- transformation model-to-model (PIM to PIM)

Generally speaking, the model-to-code transformation can be seen as a special case of model-to-model transformation. A classification of model transformation approaches is presented in [5]. Basically, a model-to-model transformation is performed with the help of transformation rules that consists in transforming a set of input models into a set of targeted models.

C. Concept of pivot model

Thank to this architecture and according to MDE, a pivot model can be considered as a PIM (level M1) and the PSM can be associated to an application specific model such as optimization model formalism. Basically, a PIM is supposed to be available initially, then PIM to PSM or PIM to PIM transformations have to be computed by applying transformation processes. Generally speaking, the PIM construction is built from elementary models, denoted EM , that describe element parts of the system. An elementary model EM , in the field of optimization, is associated with a subspace of a vector space defined on \mathbb{R}^n . It is considered that the integer set \mathbb{N} is a specialization of the real number space \mathbb{R} : $\mathbb{N} \subset \mathbb{R}$ and that the assertions *True* and *False* are modeled with binary values, respectively 1 and 0, i.e. a specialization of \mathbb{N} . An element model representing an element in a given mode is defined as:

Definition 1: $EM : \text{mode}(EM) \leftrightarrow \mathcal{V}_S \in \mathcal{E}; \mathcal{E} \subset \text{dom}(S) \subset \mathbb{R}^n$
with:

- $\mathcal{V}_S = \{v_0, \dots, v_{n-1}\}$ is a set of variables respectively related to the tuple of symbols $\mathcal{S} = \{\text{symbol}_0, \dots, \text{symbol}_{n-1}\}$
- $\text{dom}(S) = \text{dom}(\text{symbol}_0) \times \dots \times \text{dom}(\text{symbol}_{n-1})$ is the set of value domain of symbols corresponding to variables.
- mode is generally and implicitly ok (except in diagnosis analysis) for normal behavior
- the subspace \mathcal{E} is defined by a set of n_j constraints \mathcal{K} defined over \mathbb{R}^n .

$$\mathcal{K} = \{\mathcal{K}_j(S) \diamond 0; \forall j\} \quad (1)$$

where \diamond stands for a comparison operator.

The notion of element has to be clarified. Let's consider a dual flow ventilation system [6] composed of two speed variation control devices and two electric drives associated with the extraction of indoor air. The electric drive and the control system models are parts of the overall ventilation system model but at a lower level of consideration. The more the elements are decomposed, the more they can be reused. Actually, a pivot model is composed step by step by adding required element models.

Definition 2: A pivot system model $PM = (\mathcal{K}_\Sigma(S_\Sigma, \text{dom}(S_\Sigma)))$ is an union of elementary models EM_i plus connection constraints K_j

Let's consider the transformation of a PIM acausal model into a PSM causal i.e. simulable, model.

Consider the pivot system model $PM = (\mathcal{K}_\Sigma(S_\Sigma, \text{dom}(S_\Sigma)))$. Constraints can be decomposed into

equality constraints, denoted $\mathcal{K}_\Sigma^=$, and inequality constraints, denoted \mathcal{K}_Σ^\leq . A model is said *simulable* if it exists a function $\varphi: \mathcal{S}_\Sigma^{in} \rightarrow \mathcal{S}_\Sigma^{out}$ such as $\mathcal{K}_\Sigma^=(S_\Sigma) \leftrightarrow \varphi(S_\Sigma^{in}) = S_\Sigma^{out}$ on $\text{dom}(S_\Sigma)$ with $(S_\Sigma^{in}, S_\Sigma^{out})$ is a partition of \mathcal{S}_Σ .

Transforming PM into a simulation model for simulated annealing optimization for example, denoted PM_{SA} , consists in selecting and projecting $\mathcal{K}_\Sigma^=(S_\Sigma)$ into $\varphi(S_\Sigma^{in}) = S_\Sigma^{out}$, a causal ordering has to be performed. It requires usually to set values of some variables that will become parameters and input variables. The Dulmage-Mendelsohn algorithm [29] is generally used for this purpose.

A MILP model is defined as $PM_{MILP} = (\mathcal{K}_\Sigma^{MILP}(S_\Sigma^{MILP}), \text{dom}(S_\Sigma^{MILP}))$ with $\mathcal{K}_\Sigma^{MILP}(S_\Sigma^{MILP})$ linear. Transforming PM into PM_{MILP} relies on linearization transformation patterns.

Different transformation processes for composing a pivot model are detailed in section III, then the pivot model is projected into causal simulable model and MILP formalism.

III. TRANSFORMATION PROCESSES

This section gives an overview on different transformation principles using an illustrative example. Transformation process is composed of two main steps. The first one aims at manipulating element models to get a system pivot model. The second step consists in applying different projections to get target application models. The processes leading to either a simulable model or a MILP model are shown in this section.

A. Composition process

This sub-section focuses on how a pivot model is built. According to the definition 2, the most important step to build a system pivot model is the composition of different element models. The objective of the composition is to help the reusability of element models and to make the pivot model more modular. A composition may concern a set of element models, a set of compositions of element models or a set of compositions of compositions and so on. Moreover, recursive compositions can be performed unlimitedly to get bigger compositions. To illustrate this point, consider now an electric circuit as presented by figure 1.

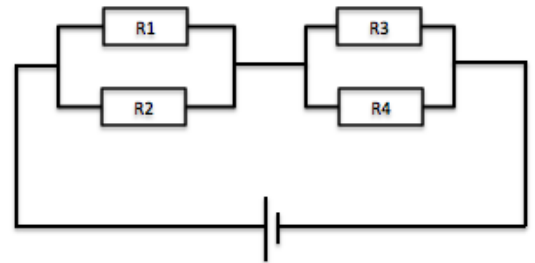


Fig. 1: Example of electric circuit

The system presented in figure 1 is composed of two blocks of 4 resistors R_1, R_2, R_3 and R_4 . Independently of any formalism, the construction of such a pivot model can be

done by composing firstly a bloc of 2 parallel resistors. Then, the pivot model is build by duplicating this bloc and connecting the whole system.

When composing step by step the pivot model, there are two remaining problems that have to be considered. The first one consists in specializing all resistors with the corresponding values, and the second one consists in establishing the different connections between element models.

To deal with the first problem, each element models (*EM*) is necessary specialized before being used in a composition. The specialization concept presented in [34] is well suited for this problem. It makes an element more specific by adding some additional information like a prefix or a type. According to the definition 1, the specialization of an *EM* consists only in adding a distinct prefix to symbol representing a variable each time it is used. For instance, $R_1.U$ is not like $R_2.U$ and so on. An *EM* could be specialized as many times as desired. The more specialized an *EM* is during a composition process, the more specific it is. For instance, $bloc_1.R_1.U$ is not like $bloc_2.R_1.U$. Nevertheless, a set of specialized *EM* cannot form a composition without connections between them. Indeed, two specialized *EM*, for instance resistors R_1 and R_2 require explicitly the following connecting-equations, which is a common concept with [34]:

$$\begin{aligned} R_1.U &= R_2.U \\ I_{total} &= \frac{R_1.U}{R_1}.R + \frac{R_2.U}{R_2}.R \end{aligned}$$

These connecting equations are added into the compositions. A pivot model for this system is given by:

- the parallel bloc composition with the dots '.' represent suffixes of prefixes to symbols standing for variables.

$$C_0 : R_1.U = R.R \times R_1.I \quad (2)$$

$$C_1 : R_2.U = R_2.R \times R_2.I \quad (3)$$

$$C_2 : R_1.U = R_2.U \quad (4)$$

$$C_3 : I_{total} = \frac{R_1.U}{R_1.R} + \frac{R_2.U}{R_2.R} \quad (5)$$

- By duplicating the parallel bloc composition above twice and by adding connecting-equations for establishing the final circuit. The system pivot model is thus built:

$$C_0 : bloc_1.R_1.U = bloc_1.R_1.R \times bloc_1.R_1.I \quad (6)$$

$$C_1 : bloc_1.R_2.U = bloc_1.R_2.R \times bloc_1.R_2.I \quad (7)$$

$$C_2 : bloc_1.R_1.U = bloc_1.R_2.U \quad (8)$$

$$C_3 : bloc_1.I_{total} = \frac{bloc_1.R_1.U}{bloc_1.R_1.R} + \frac{bloc_1.R_2.U}{bloc_1.R_2.R} \quad (9)$$

$$C_4 : bloc_2.R_1.U = bloc_2.R_1.R \times bloc_2.R_1.I \quad (10)$$

$$C_5 : bloc_2.R_2.U = bloc_2.R_2.R \times bloc_2.R_2.I \quad (11)$$

$$C_6 : bloc_2.R_1.U = bloc_2.R_2.U \quad (12)$$

$$C_7 : bloc_2.I_{total} = \frac{bloc_2.R_1.U}{bloc_2.R_1.R} + \frac{bloc_2.R_2.U}{bloc_2.R_2.R} \quad (13)$$

$$C_8 : bloc_1.I_{total} = bloc_2.I_{total} \quad (14)$$

$$C_9 : U_{total} = bloc_1.R_1.U + bloc_2.R_1.U \quad (15)$$

To summarize the above pivot model construction, the resistor model is firstly specialized twice to create two different resistors. Then, a bloc of two parallel resistors is created by adding connecting equations. Finally, the pivot model is built by duplicating this parallel bloc and adding new connecting equations. This pivot model can automatically be generated if these three steps are defined in a recipe. The concept of recipe is an important tool for the systematic generation and transformation processes.

Each generation or transformation step is considered as a transformation rule, which is implemented and put into a common rule-set. Then, recipes have then to be developed: they trigger rules from a rule-set in a relevant order until a desired formalism is obtained for a given application. However, the equation manipulation to create such a pivot model is not a trivial task. To automatize the addition of prefixes, the combinations with connecting constraints, the constraint time duplication, an symbolic calculation engine is required.

In the recent decades, symbolic computation or computer algebra [35, 37] have become an important research area of mathematics and computer sciences aiming at developing tools for solving symbolic equations. The capabilities of major general purpose Computer Algebra Systems (CAS) is presented in [36, 38]. Moreover, among the mathematical features of a CAS, there are transformations allowing to manipulate and optimize symbolic computations in order to automatically generate optimization code [39].

For instance, the GIAC/XCAS CAS [40] has been developed to solve a wide variety of symbolic problems and has been awarded with the 3rd price at the Trophées du Libre 2007 in the scientific software category [41]. This CAS has been used for manipulation of symbols in all the constraints of the pivot model. With GIAC/XCAS, each constraint is considered as a n-ary equation tree as presented in figure 2.

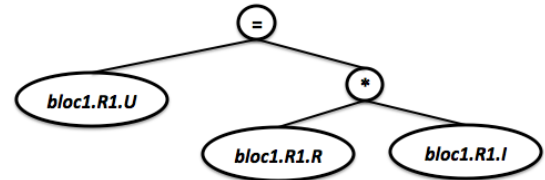


Fig. 2: n-ary tree representation for equation (6)

Finally, the set of required manipulations for composing a pivot model is respectively summarized as follows:

- specialization of *EM* by adding prefixes
- addition of connecting-equations

B. Projection process

Once a pivot model is composed, the next step consists in applying different projection processes to get desired formalism. These projection processes can always be detailed in recipes to automatize the transformation between models. This sub-section shows different steps to get causal simulable model and energy management model formalism. These processes are summarized in figure 3.

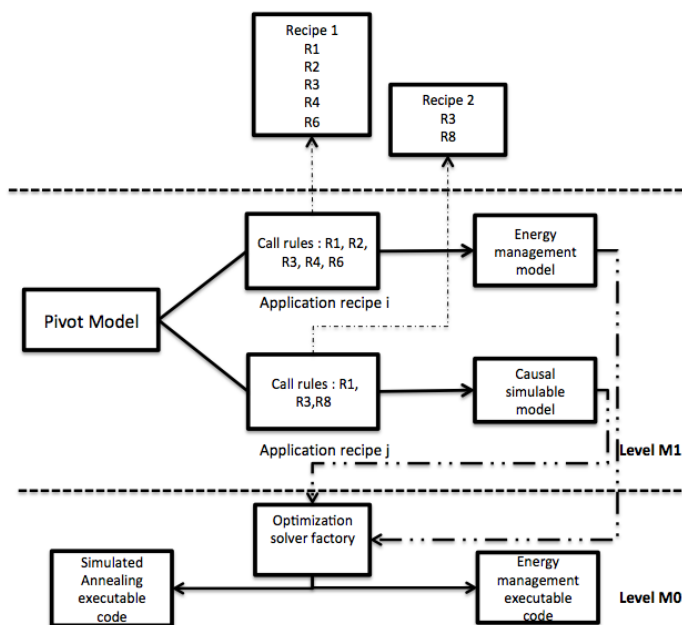


Fig. 3: Example of two projection processes

Energy management application using MILP formalism requires an acausal linear model containing 3 kinds of variables : binary, continuous, integer and 2 kinds of constraints: equality and inequality. It means that MILP transformation process aims at transforming and linearizing all constraints into equality and inequality constraints.

Based on the platform PREDIS/MHI model detailed in section IV, the first point is necessary performed to transform all ODE and logical constraints into equality and inequality constraints. In this study case, an approximation of ODE time discretization is shown instead of the exact transformation solution. This approximation consists in developing the derivative variable into:

$$\frac{dv_i}{dt} = \frac{v_i(t+1) - v_i(t)}{t} \quad (16)$$

with $v_i \in \mathcal{V}_S$ and t is the pre-defined time step. According to the obtained results, it is considered as precise enough for the 1 hour time step of the energy management.

This time discretization transformation of all ODE, $\frac{dv_i}{dt} = f(\mathcal{V}_S)$, is performed symbolically as presented by figure 4.

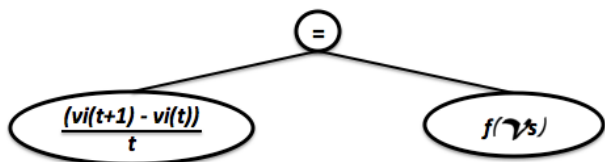


Fig. 4: Time discretization pattern

The main idea of this transformation is the same for logical constraint transformation and it can be found in [12]. Once this step is completed, the pivot model contains equality and inequality constraints. The next step to do consists in searching and linearizing all non-linear terms.

The difference between non-linear terms is based on the nature of variables and/or the nature of functions that contain variables. Indeed, product of two discrete variables cannot be linearized in the same way as a product of two continuous variables or a cosine function for example. To linearize the pivot model, it is preferable to sort out all the non-linear terms in different kinds of non-linearity first. Then each kind of non-linearity is linearized by corresponding rules. It means that recipes, rules and rule-set have to be easily extended to cover all possible changes. The whole linearization process is summarized in figure 5.

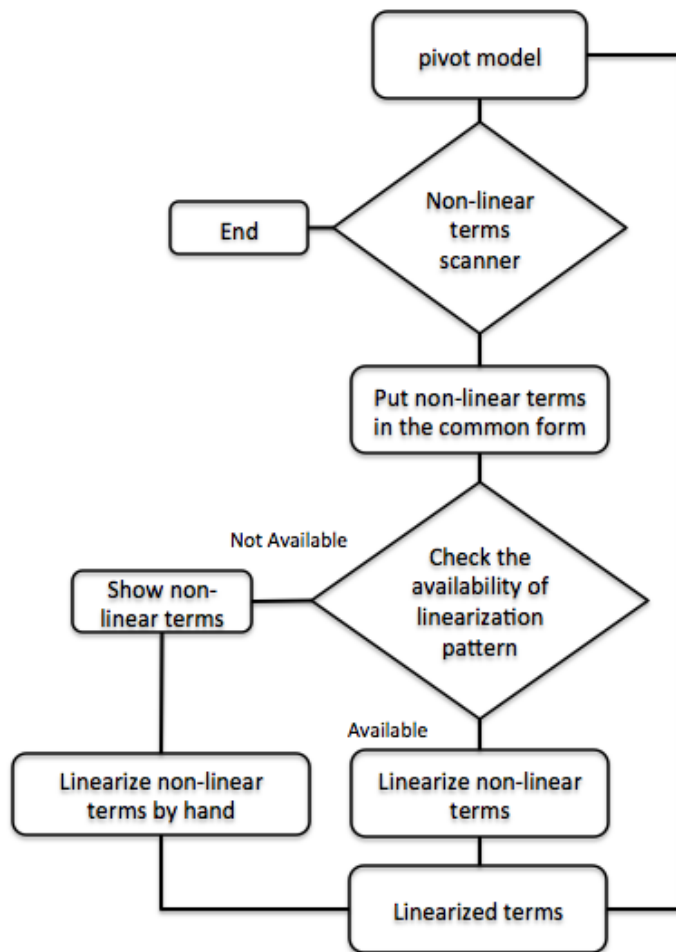


Fig. 5: Linearization process

This schema shows how the linearization process can be automatized using different patterns that were presented in [12, 42]. This process deals with non-linear terms as follows:

- product of m binary variables with $m \geq 1$
- product of l discrete variables with $l \geq 1$
- product of m binary variables and l discrete variables
- product of m binary variables and 1 continuous variable
- product of l discrete variables and 1 continuous variable
- product of m discrete variables and l discrete variables and 1 continuous variable

However, there are some terms for which the linearization process cannot be automatized and where a human intervention

is required, for instance the product of n ($n \geq 1$) continuous variables. Indeed, it does not exist a linearization pattern for this type of non-linearity to be performed directly. Linearizing a such of non-linearity requires a preliminary step consisting in discretizing the domain of $n - 1$ continuous variables into sets of discrete values. Then, the pattern of discrete and continuous product can be used to get a linear term. Discretization also means approximation to realistic values, therefore the choice of discrete values impacts strongly on final results and this step can not be automatically performed by system. Only expert who masters his dwelling system can take good values for linearization process afterwards.

Let's linearize the circuit system (1) by discretizing for instance the resistor into $R = \{3, 4\}$. Then the discrete and continuous linearization pattern can be used by introducing a new variable, denoted Z , representing the product $R \times I$ with:

$$Z = R \times I = (\delta_1 \times v_1 + \delta_2 \times v_2) \times I \quad (17)$$

with δ_i is a binary variable that takes value in $\{0, 1\}$. Actually, the goal is to select the best value among those of R to maximize or minimize the objective function. Equation (17) can be factorized as:

$$Z = \delta_1 \times v_1 \times I + \delta_2 \times v_2 \times I \quad (18)$$

with v_1 and v_2 standing for parameters. There are two binary and continuous products to be linearized. Let's linearize for instance the first binary and continuous product term: $\delta_1 \times v_1 \times I$. The corresponding pattern implies to create a new continuous variables, denoted Z' with 4 new constraints delimiting the bounds of Z' given by:

$$Z' \leq \delta_1 \times v_1 \times \bar{I} \quad (19)$$

$$Z' \geq \delta_1 \times v_1 \times \underline{I} \quad (20)$$

$$Z' \leq (I - \underline{I} \times (1 - \delta_1)) \times v_1 \quad (21)$$

$$Z' \geq (I - \bar{I} \times (1 - \delta_1)) \times v_1 \quad (22)$$

with \underline{I} and \bar{I} respectively are lower and upper bound of the continuous variable I . The second binary and continuous product $\delta_2 \times v_2 \times I$ is linearized in the same way. Once all the non-linear terms are linearized, the MILP model formalism is obtained.

Regarding the generation of a causal simulable model for local optimization, the required projection aims at transforming the pivot model into a simulable model. Firstly, a model is simulable if only if it is a structurally just-determined model [29] i.e. the number of variables is equal to the number of equality constraints, therefore one solution value can be assigned to each variable. Dulmage-Mendelsohn algorithm [29] has been used to compute just-determined blocks of constraints and variables which are represented by a reorganized structural matrix. However, some pivot models can also be:

- structurally under-determined i.e. they contain less equality constraints than variables. In this case, variables will have an infinite number of possible solutions yielding non-simulable models. Nevertheless, non-simulable models are frequent in energy management optimization applications.

- structurally over-determined i.e. there are more equality constraints than variables. In this case, generally speaking, no one value can be assigned to some variables i.e. no one solution can be computed. This kind of models are nevertheless common in diagnosis application where no solution means failure.

Normally, a correct simulable model gives only a just-determined set while other sets are empty. The equality constraints can be reorganized according to the upper-triangular just-determined part of the incidence matrix of equality constraints. The presence of an under-determined set or of an over-determined part means that the whole model can't be simulated and it is necessary to recheck the element models.

In order to automatize the whole transformation process, a software architecture is proposed in figure 6.

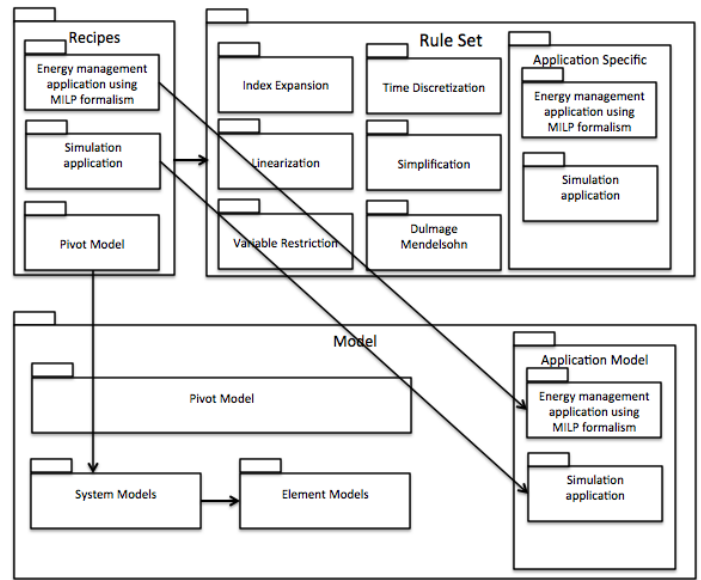


Fig. 6: software architecture

The key steps to transform a pivot model into energy management using MILP formalism and simulation are shown. The next section illustrates the application of the proposed method to the model transformation of the PREDIS/ENSE3 platform.

IV. APPLICATION EXAMPLE

This section presents the platform PREDIS/MHI located in Grenoble, France, that will be used as a "fil-conducteur" to explain the proposed approach. The Monitoring and Habitat Intelligent PREDIS platform is a research platform dedicated to research about smart-building for company, academic researchers and students.

This platform is a low consumption office building highly instrumented where most of the energy flows are measured using different sensor technologies. The structure of this platform is given by figure 7. For the sake of clarity, this section focuses on the classroom zone that is equipped with computers for students and a heating and dual flow ventilation system containing:

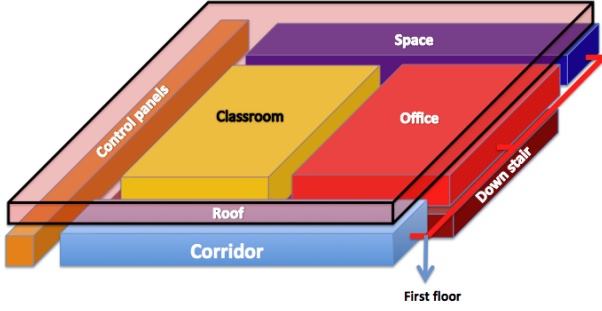


Fig. 7: Overview of PREDIS/ENSE3 platform

- an air treatment unit model:

$$AirFlow = coef \times Q_{Air} \quad (23)$$

$$P_{airTreatmentUnit} = P_{ventilation} + P_{heating} \quad (24)$$

- a thermal balance model:

$$Phi_{Total} = Phi_{Sun} + P_{heating} + Phi_{Occup} \quad (25)$$

- a thermal comfort model depending on whether there is someone or not in the classroom:

$$If\ presence = 1, \quad (26)$$

$$T_{felt} < T_{pref} \Rightarrow sigma_{incomfort} = 1/(T_{pref} - T_{max}) \times T_{felt} - T_{pref}/(T_{pref} - T_{max})$$

$$T_{felt} \geq T_{pref} \Rightarrow sigma_{incomfort} = 1/(T_{max} - T_{pref}) \times T_{felt} - T_{pref}/(T_{max} - T_{pref}) \quad (27)$$

$$If\ presence = 0, sigma_{incomfort} = 0 \quad (28)$$

$$T_{felt} \leq T_{max,absence}$$

$$T_{felt} \geq T_{min,absence}$$

- a Thermal Zone model:

$$R_{ventilation} = 1/((1 - efficiency) \times Cp_{Air} \times rho_{Air} \times AirFlow) \quad (29)$$

$$R_{Eq} = 1/(1/(R_{ventilation} + R_w) + \sum(1/R[neighborhood])) \quad (30)$$

$$\frac{d}{dt} T_w = -1/(R_{Eq} \times C_w) \times T_w + 1/((R_{ventilation} + R_w) \times C_w) \times T_{out} + \sum(T[neighborhood]/(R[neighborhood] \times C_w)) + R_{ventilation} \times Phi_{total}/(C_w \times (R_{ventilation} + R_w)) \quad (31)$$

$$T_{In} = R_{ventilation} \times T_w / (R_{ventilation} + R_w) + R_w / (R_{ventilation} + R_w) \times T_{out} + R_{ventilation} \times R_{Eq} \times Phi_{total} / (R_{ventilation} + R_w) \quad (32)$$

- a CO₂ Comfort model:

$$sigma_{CO_2} = (C_{CO_2} - C_{fav}) / (C_{max} - C_{fav}) \quad (33)$$

- a CO₂ Zone model:

$$\frac{d}{dt} C_{InCO_2} = Q_{Breath} \times occupancy \times (C_{Breath} - C_{InCO_2}) / Vol_{Zone} + AirFlow \times (C_{OutCO_2} - C_{InCO_2}) / Vol_{Zone} \quad (34)$$

- and finally, the total power consumption model:

$$P_{total} = P_{airTreatmentUnit} + P_{lighting} + P_{computer} \quad (35)$$

$$Total_{cost} = P_{total} \times PricePerKwh \quad (36)$$

These models describe only the physical phenomena of PREDIS/ENSE3. This section illustrates how the pivot model of the classroom is constructed, how it is projected it into MILP formalism and into a causal simulable model. Different required steps to get these application models are summarized in the figure 8.

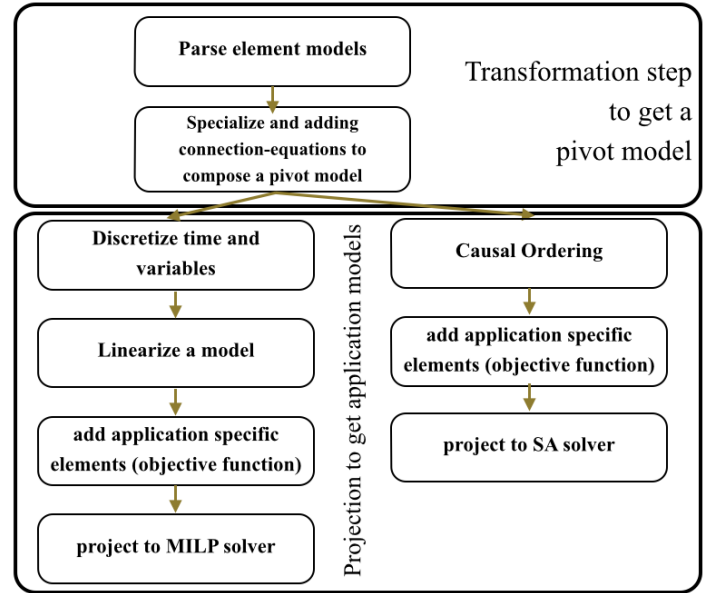


Fig. 8: Transformation processes to get target models

V. FROM PIVOT TO A SIMULABLE MODEL

To generate PREDIS/ENSE3's pivot model, the composition recipe is realized in 3 steps:

- Compose the CO₂ system:
 - specialize : CO₂ comfort with prefix : CO₂Comfort.
 - specialize : CO₂ comfort with prefix : CO₂Zone.
 - connect : CO₂Comfort.C_{CO₂} = CO₂Zone.C_{InCO₂}
- Compose the Thermal system:
 - specialize : thermal comfort with prefix : thermalComfort.
 - specialize : thermal zone with prefix : thermalZone.
 - connect : thermalComfort.T_{felt} = thermalZone.T_{In}
- Compose the final pivot model:
 - specialize : CO₂ system with prefix : CO₂System.
 - specialize : thermal system with prefix : thermalSystem.
 - specialize : power consumption with prefix : powerConsumption.
 - specialize : thermal balance with prefix : thermalBalance.
 - specialize : air treatment unit with prefix : airTreatmentUnit.
 - connect : airTreatmentUnit.AirFlow = thermalSystem.AirFlow

- connect : airTreatmentUnit.P_{airTreatmentUnit} = powerConsumption.P_{airTreatmentUnit}
- connect : thermalBalance.P_{heating} = airTreatmentUnit.P_{heating}

Initially, element models of PREDIS/ENSE3 are represented in textual description files. Thanks to the GIAC/XCAS computer algebraic system [40], each constraint is represented as a n-ary equation tree. The different variables belonging to the different constraints are collected and represented by symbols. After the parsing process, an elementary model EM is represented by a set of n-ary equation trees that facilitates the different manipulations and projection afterwards. Consider now the representation of the CO₂ zone model. It yields a n-ary equation tree given by figure 9.

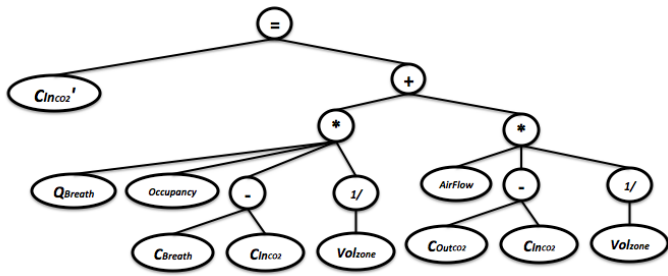


Fig. 9: CO₂ zone model n-ary representation

The name of constraints is then specialized by adding given prefixes and new connection equations are added to compose the pivot model of the system. Once these connection-equations are taken into account, the PREDIS/ENSE3's pivot model generation process is completed.

To transform the PREDIS/ENSE3's pivot model into a simulable model formalism, the key step is to perform Dulmage-Mendelsohn algorithm [29] to verify if the PREDIS/ENSE3's model could be simulable, then the causal ordering of variables if it is simulable.

In building energy management, the reorganized structural matrix is usually upper triangular with no block on the diagonal but sometimes blocks may appear. In this case, the projection cannot be fully automatized because there is no general process to solve implicit sub-systems of nonlinear equations.

It is important to note that only equality constraints are taken into account for the generation of a simulable model. It means that a preliminary step is required: the extraction of equality constraints. If the set of equality constraints is just-determined, the next step consists in making this pivot model simulable. On other words, this pivot model S_{Σ} need to be separated into S_{Σ}^{in} and S_{Σ}^{out} . Therefore, causal ordering process is necessary performed using a Dulmage-Mendelsohn based algorithm.

Let's consider a practical and didactic example consisting in simulating the thermal part of a hybrid panel running under sun:

The model of the hybrid panel is built around Hottel-Whillier equations. This equation describe phenomenon observed in the system of energy caption and transmission.

The system described is a panel of photo voltaic cells which are cooled by liquid in circulation under the layer of PV cells as shown in the scheme

$$C_0 : F_R = \frac{\phi * C_P}{(S_{PV} * U_{loss})} * (1 - e^{-S_{PV} * U_{loss} * F' / (\phi * C_P)})$$

$$C_1 : P_{Ther} = S_{PV} * F_R * \sigma_{abs} * G + U_{loss} * (T_{outdoor} - T_{Input})$$

$$C_2 : P_{Ther} = \phi * C_P * (T_{Output} - T_{Input})$$

with:

F_R : Heat dissipation factor.

ϕ : Flow in the panel.

C_P : Heat capacity.

S_{PV} : Panel Surface Area.

U_{loss} : Heat transfer coefficient.

F' : Thermal resistance between cells.

G : solar radiation.

P_{Ther} : Heat recovery capacity.

T_{Output} : Output temperature of the coolant.

T_{Input} : Input temperature of the coolant.

$T_{outdoor}$: Ambient temperature.

σ_{abs} : Thermal solar panel performance.

The variables presented above contribute to the description of the physical aspects and operational aspects of the hybrid panel. The physical aspects variables are fixed and considered as parameters for the simulation. This parameters are mandatory informed before the simulation process. The causality imposed for the simulation consist to consider the parameters as inputs of simulation and the variables, degrees of freedom, as outputs. The system is simulable if it can supply exact outputs for the inputs chosen we say that is just determined. If the simulation need more information, it is considered as under-determined. If there is one or more degrees of freedom the simulation can be considered as over-determined. The values of the parameters are integrated as equations for dulmage-mendelshon algorithm:

$$C_3 : C_P = value \quad (37)$$

$$C_4 : T_{Outdoor} = value \quad (38)$$

$$C_5 : T_{Input} = value \quad (39)$$

$$C_6 : S_{PV} = value \quad (40)$$

$$C_7 : F_R = value \quad (41)$$

$$C_8 : \sigma_{abs} = value \quad (42)$$

$$C_9 : G = value \quad (43)$$

$$C_{10} : U_{loss} = value \quad (44)$$

$$C_{11} : F_{prim} = value \quad (45)$$

The dulmage-Mendelshon algorithm check the simulability of the system as shown in the matrix I which is triangular. The demonstration of the just determination of the model.

The whole $C_3 \dots C_{11}$ must be informed to get the model simulable . The variables T_{Output} , ϕ , P_{Ther} are considered as

	T_{Output}	ϕ	F_{prim}	P_{Ther}	U_{loss}	G	σ_{abs}	F_R	S_{PV}	T_{Input}	$T_{Outdoor}$	C_P
C_2	1	1	0	1	0	0	0	0	0	1	0	1
C_0	0	1	1	0	1	0	0	1	1	0	0	1
C_{11}	0	0	1	0	0	0	0	0	0	0	0	0
C_1	0	0	0	1	1	1	1	1	1	1	1	0
C_{10}	0	0	0	0	1	0	0	0	0	0	0	0
C_9	0	0	0	0	0	1	0	0	0	0	0	0
C_8	0	0	0	0	0	0	1	0	0	0	0	0
C_7	0	0	0	0	0	0	0	1	0	0	0	0
C_6	0	0	0	0	0	0	0	0	1	0	0	0
C_5	0	0	0	0	0	0	0	0	0	1	0	0
C_4	0	0	0	0	0	0	0	0	0	0	1	0
C_3	0	0	0	0	0	0	0	0	0	0	0	1

TABLE I: Results of Dulmage-Mendelsohn decomposition

the output for simulation. The Dulmage-Mendelsohn algorithm produce the matrix I. This matrix is used to order equations in the execution process, it is the causality ordering. It consist to schedule the resolution of equations to get result from each one. The last equation (row of the matrix) C_3 has to be solved firstly. The resolution of C_3 equation results the value of C_P . The result is put in the inputs set for the next equations. The process run until the whole equations solved. The last equation to be solved is C_2 and give the last output T_{Output} . The other outputs are computed before: ϕ from C_0 and P_{Ther} from C_1 .

VI. PROJECTION TO A MILP OPTIMIZATION MODEL

To transform the PREDIS/ENSE3's pivot model into a MILP model formalism, a specific recipe is built as presented by the below part of the figure 8. Let's detail some specific rules to illustrate how the symbolic transformation is performed : time discretization and linearization.

The time discretization consists in discretizing the pivot model into 24 sampling periods standing for one day. Thus, this step multiplies 24 times each constraint of the pivot model with time index ranging from 0 to 23. The ODE implementation of CO_2 zone at 5th time step is given by figure 10.

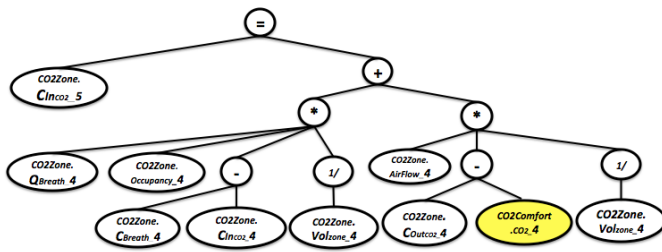


Fig. 10: CO₂ Zone model after the ODE transformation processing

The next important projection consists in linearizing non-linear terms inside the constraints of the pivot model. First, all non-linear terms are detected; then, the nature of each nonlinear term is analyzed before being recursively linearized according to the corresponding pattern according to the linearization process presented by the figure 5. For instance, the binary-continuous product : $CO_2Zone.Q_{Breath} \times CO_2Zone$ is linearized. *occupancy* in the CO_2 zone model where *occupancy* is 0 whether there is nobody or 1 otherwise. In this case, a new variable, denoted z , is used for replacing the considered term in the corresponding constraint as given by figure 11.

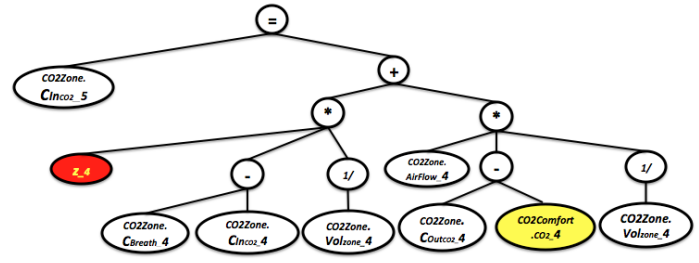


Fig. 11: CO₂ Zone binary model after the first linearization processing

And four new constraints resulting of this binary-continuous product are added into the pivot model :

$$\begin{aligned}
 z_4 &\leq CO_2Zone.occupancy_4 \times \overline{CO_2Zone.Q_{Breath_4}} \\
 z_4 &\geq CO_2Zone.occupancy_4 \times \underline{CO_2Zone.Q_{Breath_4}} \\
 z_4 &\leq \overline{CO_2Zone.Q_{Breath_4}} - (1 - CO_2Zone.occupancy_4) \times \underline{CO_2Zone.Q_{Breath_4}} \\
 z_4 &\geq \underline{CO_2Zone.Q_{Breath_4}} - (1 - CO_2Zone.occupancy_4) \times \overline{CO_2Zone.Q_{Breath_4}}
 \end{aligned}$$

Resulting of this linearization pattern represents exactly the considered binary-continuous product because:

- if $occupancy_4 = 1$:
 - $z_4 \leq \sup(CO_2Zone.Q_{Breath_4})$
 - $z_4 \geq \inf(CO_2Zone.Q_{Breath_4})$
 - $z_4 \leq \overline{CO_2Zone.Q_{Breath_4}}$
 - $z_4 \geq \underline{CO_2Zone.Q_{Breath_4}}$

In this case, the two first constraints are always true so they can be eliminated. The last two constraints make it possible to take into account the real values of $CO_2Zone.Q_{Breath_4}$.
- if $occupancy_4 = 0$:
 - $z_4 \leq 0$
 - $z_4 \geq 0$

when there is nobody in the classroom, it means that the Q_{Breath} is equal to 0, too.

After linearizing all non-linear terms, the obtained model is ready to provided to MILP solver to generate energy management plans. This MILP formalism has 1679 constraints/1129 variables whose 1011 new constraints/459 new variables added resulting of linearization process.

VII. PROJECTION TO A SIMULATED ANNEALING OPTIMIZATION MODEL

The simulated annealing process is based on stochastic approach which finds an optimum for single objective optimization problem. In this study, it is used as complement to MILP optimization process to find quickly a better solution in case of minor changes in the optimizing problem.

The aim in the development of simulated annealing optimizer is to complete the offer of MILP optimizing. MILP optimization, thanks to tools like CPLEX or GLPK, offer the guaranty of global optimum when the optimization is achieved. The issue with this approach is the computation time which is considerable in case of complex problems. The long time computation can be acceptable in case of anticipation but in real time situations this running time is not acceptable. For example, when the system interacts with user, recommend, for an ergonomic use, thirty seconds as maximum waiting time during interaction.

The Simulated annealing is not the fastest algorithm in absolute. But the interaction means changes and adaptations of an initial optimization problem. The idea in using the SA Algorithm is to take into account the results of optimizing process for an initial problem solved by Cplex, GLPK or other MILP solvers. The SA optimization takes as initial solution the one that is generated by the MILP solver for the initial problem. The new problem posed by interactions with occupant is quite different. The difference can be in the value of parameters like comfort temperature for example or additional constraints in the problem to describe a limitation like minimum of ventilation air flow because of steam cooking for example.

The simulation models used in SA has to describe the same phenomenon as for MILP. this common base is the guaranty of credibility in initialization point use.

The common description of the phenomenon is done in the pivot model, which is common for both MILP and SA optimization approaches. The recipe to transform the pivot model to simulation model is used. The SA algorithm 12 uses the model in simulation step.

The other important side of the simulated annealing optimization process is the neighborhood definition function. This function determine the direction and the jump of the value for degrees of freedom in each iteration. To illustrate the aims in definition of this function let's get a simple example. A continuous one dimension problem in term of inputs: only one degree of freedom x for the problem.

$$Obj_i = X_i^2 + X_i + 1 \tag{46}$$

A simple and universal neighborhood function is

$$X_n = X_{n-1} + rand(-1, 1) * radius \tag{47}$$

the radius is a parameter fixed for SA optimization algorithm

The optimization problem treated here is to find the minimum for the variable Obj . The results for this simple problem are shown in the convergence curve 13

The optimization algorithm used for the problem 46 is used on four parallel executions. This chose of four executions is

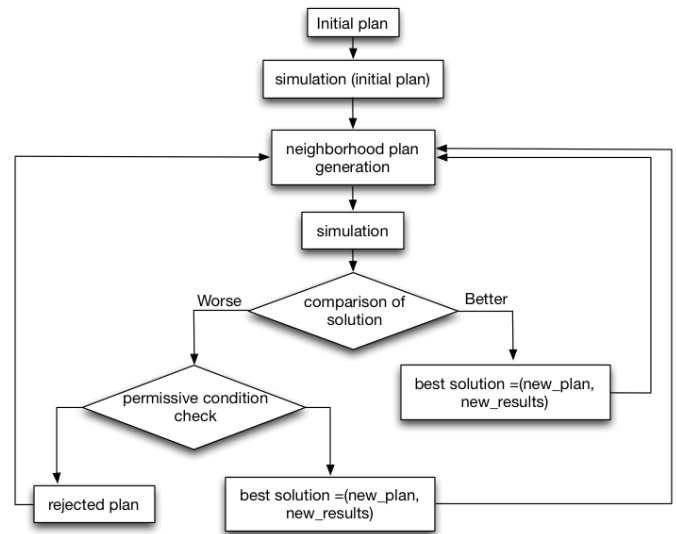


Fig. 12: Simulated Annealing Algorithm

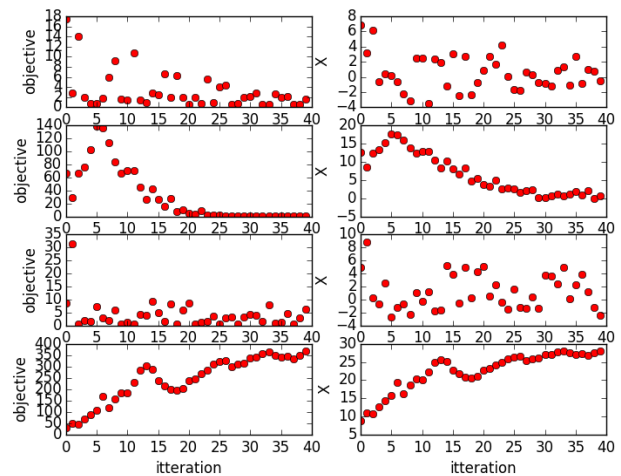


Fig. 13: optimal research evolution on four parallel processes

done because of four core processors in the computer used. It is to maximize the used of the processors during computation and to enhance chances to find best solution in the same time computation as simple execution.

The example shown elapse 00.017753 seconds with 40 iterations per process to find the best solution which is: (best plan: 1.0092122628718672, best performance: 0.5000424328936102).

The four parallel executions show that, some times, the algorithm diverges (the last process). There is no guaranty of convergence, that is why it is better to run parallel instances with limited number of iterations then to run simple instance with large number of iterations, it enhances chances to find solution.

In BEMS, the problem is more complex then this example

but it is in the same spirit. The complexity is added by the diversity of kinds of variables: continuous, discrete and binary. The multi-dimensions of the problem in BEMS are quite different than the single dimension problems treatment. In this paper we will not explain the details of treatment but we will draw major lines of the treatment procedure.

For these problems there is three main solutions:

the sequential treatment of variables The optimization process is done for each variable separately. The neighborhood is defined for the current variable to be optimized with considering the rest of variables fixed.

the global treatment of variables In this case, a new neighborhood is defined for the whole variables at each iteration. It is more random.

the clustering in the treatment of variables The clustering is a way to gather variables of the same type in term of optimization or to select variables which are physically close. The neighborhood of each vector (cluster of variables formed) is generated for each iteration in the SA running. The clusters are treated sequentially.

VIII. ANALYSIS OF OBTAINED RESULTS

After the computation with MILP solver (IBM ILOG CPLEX 12.3), the result obtained within 2 minutes for the classroom temperature and the corresponding cost for next 24 hours is given by figure 14.

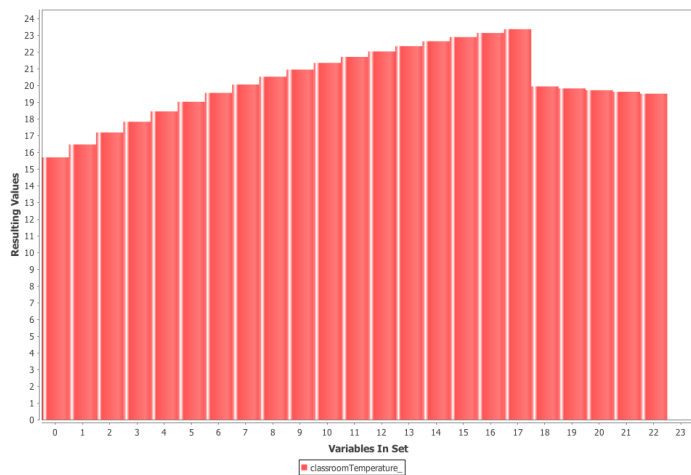


Fig. 14: Classroom temperature for 24 hours generated by MILP solver

Thanks to the pivot model, we kept the results of the MILP problem resolution and use it in the SA algorithm initializing. The problem of optimizing takes into account model of the envelop with its different faces and resistances and the HVAC system. The HVAC system is a couple of air ventilator and complex system of heating. The system of heating is water heated flowing in closed loop. In this description we consider only the power distributed by the exchangers of the heating system. An energy price model is used too. The degrees of freedom are the heating power injected by set-point temperature adjustment in the room and the air flow of the HVAC. These degrees of freedom projected in time constitute the

plan of management. The objective of the management here is quite interesting to develop. To generate the initial plan of management with cplex solver, the problem was considered with twice objectives coupled in optimizing objective. The first one are economic objective, it is to minimize the final bill during 24 hours. The second objective is comfort one, it is to maximize the occupants comfort according to comfort function description shown before. The objective for SA algorithm are exclusively economical. This difference is supposed to represent a will of a user during his interaction with the BEMS.

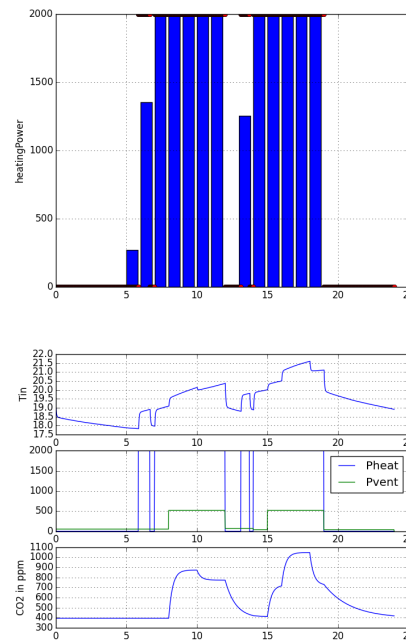


Fig. 15: initial plan with initial orders

The results presented in 15 represent the initial situation, before SA optimizing process. It is the results of CPLEX optimization with the mixed objective (economical and comfort) The results presented in 16, 17 represent two instances of SA runs. It is quite different because of the stochastic nature of the problem. The optimal solution for the problem given go toward zero for the power of heating. The optimum solution are not reached in the two tests but in the twice it was approached. The main idea for the use of SA in the BEMS is to give better solutions then the initial ones in acceptable time. This goal is more or less achieved by the results shown.

To enhance the efficiency of the algorithm and to use the maximum of computing capacities of the material, a parallelism process had been integrated. Four process run in parallel for 40 iterations per process. The elapsed time computation for this problem was 38.583835 seconds.

IX. CONCLUSIONS AND FUTUR WORKS

This paper presents a new method aiming at performing automatically model transformations to energy management system in buildings. The main contribution of this method is to avoid the model rewriting for different energy management applications that represent a significant time-consuming and

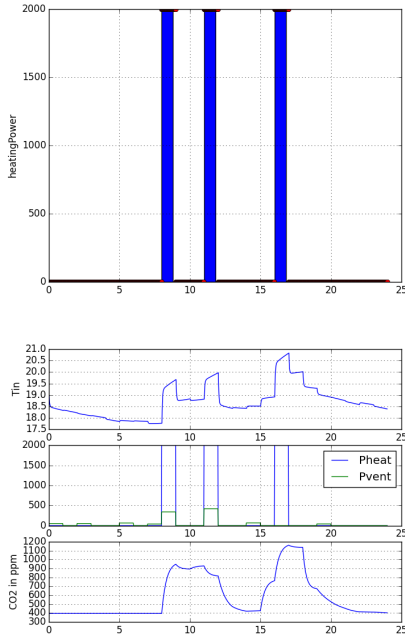


Fig. 16: SA result plan executions

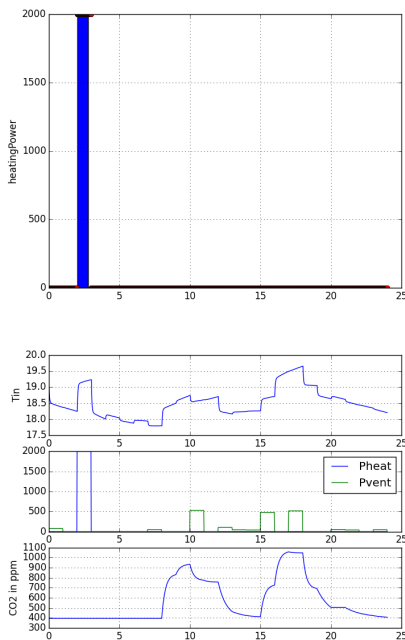


Fig. 17: SA result plan execution

error-prone. The core specifications of the construction a pivot model and the projection this last into target application models are shown via illustrative examples in order to give a better comprehension. The construction modular of pivot model encourages not only the reusability of element models but it allows to deal with equipment changes.

Based on this propose method, a software has been im-

plemented to valid this proposed method. Two kinds of target optimization models: MILP and causal simulable formalisms have been generated for the platform PREDIS/ENSE3. The first one allows generating a It is planed to extend the rule set to be able to generate results based on a global optimization approach while the second one shows acceptable results resulting of a fast heuristic local optimization approach in order to reduce time-consuming. To enhance this proposed method application field, different target applications such as : sizing, diagnosis, parameter estimation will be taken into account.

REFERENCES

REFERENCES

- [1] C. A. Balaras, A. G. Gaglia, E. Georgopoulou, S. Mirasgedis, Y. Sarafidis, and D. P. Lalas, European residential buildings and empirical assessment of the hellenic building stock, energy consumption, emissions and potential energy savings, *Building and Environment*, Volume 42, Issue 3, pp:1298-1314, 2007.
- [2] Europe's Energy Position: Markets and Supply. Market Observatory for Energy, Directorate General for Energy, report 2009
- [3] G. Levermore, *Building Energy Management Systems: An Application to Heating, Natural Ventilation, Lighting and Occupant Satisfaction*, Taylor & Francis, 2002.
- [4] S. Kent, Model driven engineering, in *Integrated Formal Methods*, ser. Lecture Notes in Computer Science, M. Butler, L. Petre, and K. Sere, Eds. Springer Berlin Heidelberg, Volume 2335, pp:286298, 2002.
- [5] P. Harmon, Mda: an idea whose time has come, Technical report, 2003.
- [6] H.A. Dang, B. Delinchant, S. Gaaloul, F. Wurtz, Electrical performance optimization of an HVAC dual flow, *Electrimacs 2011*, Cergy-Pontoise, France, 2011.
- [7] H. Doukas, K. D. Patlitzianas, K. Iatropoulos and J. Psarras, Intelligent building energy management system using rule sets, *Building and Environment*, Volume 42, Issue 10, pp:3562-3569, October 2007.
- [8] PSE. gPROMS. <http://www.psenterprise.com/gproms/>, 2011.
- [9] M. R. Bussieck, A. Meeraus, General algebraic modeling system (GAMS), In *Modeling languages in mathematical optimization*, pp:137-157, 2004.
- [10] Europe's Energy Position: Markets and Supply. Market Observatory for Energy, Directorate General for Energy, report 2009.
- [11] S.A. Ahmadi, I. Shames, F. Scotton, L. Huang; H. Sandberg, K.H. Johansson and B. Wahlberg, Towards more Efficient Building Energy Management Systems, Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on , pp:118-125, 2012.
- [12] D. L. Ha, H. Joumaa, S. Ploix, and M. Jacomino, An optimal approach for electrical management problem in dwellings, *Energy and Buildings*, Volume 45, Issue 0, pp:114, 2012.
- [13] C.A. Floudas, and P.M. Pardalos, *Encyclopedia of Optimization*, Springer Reference, 2008
- [14] S. S. Rao, *Engineering Optimization: Theory and Practice*, Third Edition, Wiley, 1996.

- [15] M. Grotschel and L. Lovász, Combinatorial optimization, Chapter 28 in R.L. Graham, Handbook of Combinatorics, Elsevier, Volume 2, pp:1541-1597, 1995.
- [16] C. H. Papadimitriou and K. Steiglitz, Combinatorial optimization: algorithms and complexity. Courier Dover, 1998.
- [17] C. Blum and A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Computing Surveys, pp:268-308, 2003
- [18] M. R. Garey and D. S. Johnson, Computers and Intractability: a guide to the theory of NP-completeness, W. H. Freeman, 1979.
- [19] I. E. Grossmann and Z. Kravanja, Mixed-integer nonlinear programming techniques for process systems engineering, Computers & Chemical Engineering, Volume 19, Supplement 1, pp:189-204, 1995.
- [20] E. L. Lawler and D. E. Wood, Branch-and-bound methods: A survey, Operations research 14.4, pp:699-719, 1966.
- [21] R. E. Barber and H. C. Lucas, System response time operator productivity, and job satisfaction, Communications of the ACM, pp:972-986, 1983.
- [22] B. Shneiderman, Response time and display rate in human performance with computers, ACM Computing Surveys 16.3, pp:265-285, 1984.
- [23] L. Jan, Impact of system response time on state anxiety, Communications of the ACM, pp:342-347, 1988.
- [24] V. Laarhoven, J. M. Peter and E. H. L. Aarts, Simulated annealing. Springer Netherlands, 1987.
- [25] H. Szu and R. Hartley, Fast simulated annealing, Physics Letters A, Volume 122, Issues 34, pp:157-162, 1987.
- [26] H. L. Minh, S. Ploix, M. Jacomino, D.L. Ha, A mixed integer programming formulation of the home energy management problem, Energy Management, INTECH, 2010.
- [27] S. G. Chouikh, Interopérabilité basée sur les standards Modelica et composant logiciel pour la simulation énergétique des systèmes de bâtiment, PhD Thesis, University of Grenoble, France, 2012.
- [28] M. Blanke, C. W. Frei, F. Kraus, R. J. Patton, M. Staroswiecki, What is fault-tolerant control?, Aalborg University, Department of Control Engineering, 2000.
- [29] A. L. Dulmage and N. S. Mendelsohn, Coverings of bipartite graphs, Canadian Journal of Mathematics, Volume 10, pp:517-534, 1958.
- [30] J. S. Ulmer, J. P. Belaud, and J. M. Le Lann, Towards a pivotal - based approach for business process alignment, International Journal of Computer Integrated Manufacturing, Volume 24, no. 11, pp:1010-1021, 2011.
- [31] J. Bézivin, On the unification power of models, Software & Systems Modeling, Volume 4, no. 2, pp:171-188, 2005.
- [32] Q. Omg, Meta object facility (mof) 2.0 query/view/-transformation specification, Final Adopted Specification (November 2005), 2008.
- [33] K. Czarnecki and S. Helsen, Classification of model transformation approaches, Proceeding of the 2nd OOP-SLA Workshop on Generative Techniques in the Context of the Model Driven Architecture, Volume 45, pp:1-17, 2003.
- [34] Modelica, Modelica® - A Unified Object-Oriented Language for Physical Systems Modeling, Technical report, Volume 3.2, 2010.
- [35] J. Von Zur Gathen, J. Gerhard, Modern Computer Algebra, Cambridge University Press, 2003.
- [36] W.S. Brown, A.C. Hearn, Applications of symbolic algebraic computation, Computer Physics Communications, Volume 17, Issues 12, pp:207-215, 1979.
- [37] F. Winkler, Advances and Problems in Algebraic Computation, in Contributions to General Algebra 12 (Proc. AAA'58, Vienna), Verlag Johannes Heyn, Klagenfurt (Austria), 2000.
- [38] M. Wester, A review of CAS mathematical capabilities, Computer Algebra Nederland, Volume 13, pp:41-48, 1994.
- [39] A. Dall'Osso, Computer algebra systems as mathematical optimizing compilers, Science of Computer Programming, Volume 59, Issue 3, pp:250-273, 2006.
- [40] P. Bernard, Giac/Xcas, a free computer algebra system. Technical report, University of Grenoble, 2008.
- [41] <http://www-fourier.ujf-grenoble.fr/parisse/giac.html>
- [42] G. D. Oliveira, M. Jacomino, D. L. Ha and S. Ploix, Optimal power control for smart homes, 18th IFAC World Congress, Elsevier, 2011.
- [43] Y. Hadjsaid, B. Lechat, C. Latremoliere and S. Ploix, Generating global energy management problems: application to the CANOPEA prototype, Building Simulation (BS2013), IBPSA World Conference, Chambéry, France, 2013.
- [44] A. Le Mounier, B. Delinchant and S. Ploix. Determination of relevant model structures for self-learning energy management system. In Building Simulation and Optimization 2014, London, U.K., 23-24 June 2014.
- [45] S. Sarabi, S. Ploix, H. Minh Le, H.-A. Dang and F. Wurtz. Assessing the relevance of reduced order models for building envelop. In Building Simulation (BS 2013), 2013.
- [46] S. Gaaloul, B. Delinchant, F. Wurtz and F. Verdire. Software components for dynamic building simulation. In Proceedings of Building Simulation 2011: 12th Conference of IBPSA 2011, Sydney, Australie, 14-16 November 2011.
- [47] S. Gaaloul, X. Hoa Binh Le, B. Delinchant, F. Wurtz and S. Ploix. Software component architecture for co-simulation applied to the coupling between a building's thermal envelope and its inhabitant behaviour. In the 38th Annual Conference of IEEE Industrial Electronics (IECON), pp: 4620 - 4625, Montréal, Canada, 25-28 October 2012.
- [48] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi Optimization by Simulated Annealing. In Science, New Series, Vol.220, No 4598 May 1983
- [49] A. Basu, LN Frazer Rapid determination of the critical temperature in simulated annealing inversion. In Science 1990